

2014

Instituto Politécnico de Coimbra
INSTITUTO SUPERIOR DE ENGENHARIA DE COIMBRA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO DE OLIVEIRA DO HOSPITAL

e-Diagnostic Software, aplicação web para monitorização de sensores em unidades industriais

MESTRADO EM COMÉRCIO ELETRÓNICO

AUTOR | Eduardo Augusto Freire Guerra Cardoso

ORIENTADOR | Prof. Doutor João António Pereira Almeida Durães
CO-ORIENTADORES | Eng.º Nuno Filipe Barbosa Varandas
| Eng.º Paulo Miguel Rocha Falcão

Coimbra, Dezembro 2014

Departamento
de Engenharia Informática e Sistemas

***e-Diagnostic Software – Aplicação web para
monitorização de sensores em unidades
industriais***

Dissertação apresentada para a obtenção do grau de Mestre em
Comércio Eletrónico

Autor

Eduardo Augusto Freire Guerra Cardoso

Orientador

Prof. Doutor João António Pereira Almeida Durães

Instituto Superior de Engenharia de Coimbra

Co-orientadores

Eng.º Nuno Filipe Barbosa Varandas

eneida wireless & sensors

Eng.º Paulo Miguel Rocha Falcão

eneida wireless & sensors

Coimbra, Dezembro, 2014

AGRADECIMENTOS

Este espaço é dedicado a todos aqueles que, de algum modo, deram o seu contributo para que este projeto fosse realizado.

Em primeiro lugar, agradeço ao meu orientador do Departamento de Engenharia Informática e de Sistemas do Instituto Superior de Engenharia de Coimbra, Professor Doutor João Durães, a forma como orientou o meu trabalho. As notas dominantes da sua orientação foram a utilidade das suas recomendações e a cordialidade com que sempre me recebeu. A liberdade de ação proporcionada permitiu desenvolver conhecimentos, que virão a demonstrar-se um grande contributo para o meu desenvolvimento pessoal e académico.

Não menos importante é deixar uma palavra de agradecimento a todos os meus professores do Mestrado em Comércio Eletrónico do Instituto Superior de Engenharia de Coimbra e da Escola Superior de Tecnologias e Gestão de Oliveira do Hospital, que também deram o seu contributo para o desenvolvimento deste projeto.

Em segundo lugar agradeço aos meus co-orientadores, Eng.º Nuno Varandas e Eng.º Paulo Falcão, pelo tempo de que dispuseram para explicar o funcionamento do atual sistema e por todas as informações necessárias referentes ao projeto proposto. Gostaria ainda de deixar um agradecimento geral a todos colaboradores da enxada que, de forma direta e indireta, puderam prestar o seu apoio.

São também dignos de uma nota de apreço neste espaço os colegas, amigos, em especial Marco Ramos, que sempre deu a sua opinião técnica e crítica para o sucesso deste projeto ao longo dos meses, aos meus familiares, que me acompanharam ao longo deste projeto, em especial ao meu tio Pedro e à minha tia Adelina, e um agradecimento especial à minha mãe Laurinda Freire e ao meu irmão Nuno Cardoso, que sempre me apoiaram em todas as etapas da minha vida, e à Marta Neves, por todo o carinho, apoio e força que me dedicou ao longo destes anos.

RESUMO

O relatório do projeto “*e-Diagnostic Software* - aplicação *web* para monitorização de sensores em unidades industriais”, respeitante ao Mestrado em Comércio Eletrónico (MCE), insere-se nas disciplinas de “Projeto ou Estágio I e II”, referentes aos primeiros e segundo semestres.

Este projeto tem como finalidade a análise e monitorização das redes de sensores instaladas nas unidades industriais. Através dos dados recolhidos pelas redes de sensores, existe a possibilidade do tratamento destes dados para controlo e monitorização sobre os processos produtivos de modo contínuo existentes nestas unidades industriais.

A oportunidade de elaboração deste projeto surge através da parceria com a empresa *eneida wireless & sensors*¹. Esta empresa é especialista em engenharia de instrumentação e comunicação, dedicando-se essencialmente ao ramo industrial, com inúmeras prestações desde a eletricidade e eletrónica industrial, automação e instrumentação à comunicação em tempo real através de infraestruturas *wireless* fiáveis e robustas. A *eneidaws* produz o seu próprio *hardware*, para projetar e instalar sistemas que permitem, de uma forma simples e económica, a recolha de informação sobre os processos produtivos de modo contínuo.

A *eneidaws* tem o seu próprio *software*, para análise, monitorização e tratamento dos dados recolhidos pertencentes às redes de sensores. No entanto, o atual *software*, *Diagnostic Software 1 (DS1)*, aplicação com funcionamento local em sistemas operativos *Windows*, contém lacunas relativamente a funcionalidades e opções pretendidas pelos clientes da *eneidaws*.

O aluno, enquanto funcionário da empresa *eneidaws* e finalista do MCE, teve como motivação apresentar este projeto na sua tese final, propondo à empresa melhorar o sistema atual para monitorização das redes de sensores, tendo em conta as necessidades da presente aplicação *DS1*, para uma nova aplicação *web* genérica multiplataforma, contemplando novas tecnologias, com o objetivo de oferecer mais funcionalidades e opções aos clientes da *eneidaws*.

A *eneidaws* propôs a integração do caso de estudo que tem atualmente neste projeto, o qual tem como finalidade monitorizar os postos de transformação energéticos, a nível ambiental, energético, de segurança, de qualidade do serviço, entre outros. Com a integração deste caso de estudo real torna-se possível validar, no final do projeto, se os objetivos definidos para a elaboração do mesmo foram atingidos.

Palavras-chave: Aplicação *web*; Monitorização; Rede de Sensores;

¹ *eneida Wireless & Sensors*. Disponível no endereço: <http://eneidaws.com/>. No restante relatório, para a referência a esta empresa será usado o termo *eneidaws*.

ABSTRACT

The project report "e-Diagnostic Software - web application for monitoring of sensors in industrial units" concerning the Master's degree in Electronic Commerce (MCE) falls within the subjects "Project or Traineeship I and II", of the first and second semesters.

This project aims at analyzing and monitoring sensor networks installed in industrial units. Through the data collected by these sensor networks, the processing of such data for continuous control and monitoring of production processes existent in these industrial units is possible.

The opportunity for the development of this project arose through a partnership with the company *eneida wireless & sensors*². This company specializes in instrumentation and communication engineering, engaged primarily in the industrial sector, with numerous services concerning electricity and industrial electronics, automation and instrumentation for real-time communication through reliable and robust wireless infrastructures. Proprietary hardware is produced by *eneidaws* to project and install systems that allow continuous gathering of information on production processes in a simple and economic manner.

Software developed by *eneidaws* permits analysis, monitoring and processing of data collected from sensor networks. However, the current software, Diagnostic Software 1 (DS1), an application run locally on Windows operating systems, has shortcomings regarding functionality and options sought by *eneidaws* customers.

The student, as an employee of *eneidaws* and MCE finalist, was motivated to present this project in his final thesis, by proposing an improvement of the existing system for monitoring of sensor networks, taking into account the needs of the current DS1 application, into a multiplatform generic web architecture, contemplating new advanced technologies, aiming to offer more functions and options to *eneidaws* customers.

The incorporation of an existing case study (aiming to monitor the transformation stations at an environmental, energetic, safety, and quality of service level, among others) into this project was proposed by *eneidaws*. The integration of the case study makes it possible to validate at the end of the project whether the objectives defined for the elaboration thereof have been achieved.

Keywords: Web Application; Monitoring; Network Sensors;

² *eneida Wireless & Sensors*. Available at: <http://eneidaws.com/>. Henceforth in the report this company shall be referred to as *eneidaws*.

ÍNDICE

1. INTRODUÇÃO	1
1.1. Enquadramento	1
1.2. Entidades Envolvidas	3
1.2.1. ISEC	3
1.2.2. ESTGOH	3
1.2.3. <i>eneida wireless & sensors</i>	4
1.3. Objetivos	5
1.4. Metodologia e Planeamento de Tarefas	6
1.4.1. Calendarização de Tarefas	6
1.4.2. Descrição das Tarefas	7
1.5. Estrutura do Relatório	9
2. CONTEXTO E DOMÍNIO DA APLICAÇÃO	11
2.1. Descrição do Sistema Atual	11
2.2. Limitações das Aplicações no Sistema Atual	14
2.2.1. Limitações da Aplicação <i>SCADA DSI</i>	14
2.2.2. Limitações da Aplicação Servidor	16
2.3. Requisitos Iniciais da Empresa	17
3. ESTADO DA ARTE	19
3.1. O que são Sistemas <i>SCADA</i>	19
3.2. Exemplos de Aplicações <i>SCADA</i>	22
3.3. Análise Comparativa dos Sistemas <i>SCADA</i>	28
3.3.1. Análise Comparativa do Sistema <i>SCADA</i> atual da <i>eneidaws</i>	29
4. ANÁLISE DE REQUISITOS E SOLUÇÃO PROPOSTA	31
4.1. Descrição da Solução Proposta	31
4.2. Requisitos Finais Para Novo Sistema	32
4.3. Atores	32
4.4. Requisitos da Aplicação Servidor	33
4.4.1. Requisitos Funcionais	33
4.4.2. Diagrama de Casos de Uso	35

4.4.3.	Requisitos Não Funcionais	36
4.5.	Requisitos da Aplicação Serviço de Alarmes	36
4.5.1.	Requisitos Funcionais	37
4.5.2.	Diagrama de Casos de Uso	37
4.5.3.	Requisitos Não Funcionais	38
4.6.	Requisitos da Aplicação <i>Web Service</i>	38
4.6.1.	Requisitos Funcionais	39
4.6.2.	Diagrama de Casos de Uso	40
4.6.3.	Requisitos Não Funcionais	41
4.7.	Requisitos da Aplicação <i>SCADA Web</i>	42
4.7.1.	Requisitos Funcionais	43
4.7.2.	Diagramas de Casos de Uso	45
4.7.3.	Requisitos Não Funcionais	47
4.8.	Vantagens Previstas do Novo Sistema	47
4.9.	Protótipos Definidos para as Aplicações	51
5.	ENQUADRAMENTO TEÓRICO DA ARQUITETURA	53
5.1.	Descrição Geral do Sistema	53
5.2.	Arquitetura Teórica das Aplicações	54
5.2.1.	Estrutura da Base de Dados	54
5.2.2.	Arquitetura da Aplicação Servidor e da Aplicação Serviço de Alarmes	56
5.2.3.	Arquitetura da Aplicação <i>Web Service</i>	58
5.2.4.	Arquitetura da Aplicação <i>SCADA Web</i>	59
5.3.	Ferramentas e Tecnologias Utilizadas	61
6.	GESTÃO DE DADOS	67
6.1.	Modelo de Dados	67
6.1.1.	Entidades Base de Dados Geral	68
6.1.2.	Relacionamentos Base de Dados Geral	69
6.1.3.	Entidades Base de Dados Rede Sensores dos Clientes	71
6.1.4.	Relacionamentos Base de Dados Rede Sensores dos Clientes	73
7.	MÓDULOS DO SISTEMA	79
7.1.	Módulo Aplicação Servidor	79

7.1.1.	Estrutura da Aplicação	79
7.1.2.	Detalhes de Implementação	81
7.1.3.	Aplicação Servidor	84
7.2.	Módulo Aplicação Serviço de Alarmes.....	85
7.2.1.	Estrutura da Aplicação	85
7.2.2.	Detalhes de Implementação	87
7.2.3.	Aplicação Serviço de Alarmes.....	89
7.3.	Módulo aplicação <i>Web Service</i>	90
7.3.1.	Estrutura da Aplicação	90
7.3.2.	Detalhes de Implementação	91
7.3.3.	Pedido <i>HTTP</i> Aplicação <i>Web Service</i>	93
7.4.	Módulo Aplicação <i>SCADA Web</i>	94
7.4.1.	Estrutura da Aplicação	94
7.4.2.	Detalhes de Implementação	97
7.4.3.	Aplicação <i>SCADA Web</i>	101
8.	CASO DE ESTUDO DA EMPRESA.....	103
8.1.	Objetivo do Caso de Estudo	103
8.2.	Enquadramento do Caso de Estudo	103
8.2.1.	Posto de Transformação.....	104
8.2.2.	<i>Hardware</i> produzido pela <i>eneidaws</i>	105
8.2.3.	Rede instalada no Posto de Transformação	108
8.2.4.	Aplicação <i>SCADA DSI</i> usada no caso de estudo	111
8.3.	Integração do Caso de Estudo no Projeto desenvolvido.....	112
8.3.1.	Cenário Real dos Sensores Instalados no PT	112
8.3.2.	Integração do Cenário na Aplicação <i>SCADA Web</i>	113
9.	ANÁLISE CRÍTICA.....	117
9.1.	Novo Sistema <i>SCADA</i>	117
9.2.	Cenários da aplicação do novo sistema <i>SCADA</i>	119
9.2.1.	Cenário 1 <i>Online</i> – Empresa de Pequena Dimensão	120
9.2.2.	Cenário 2 <i>Online</i> – Empresa de Grande Dimensão	121
9.2.3.	Cenário 3 <i>Intranet</i> – Empresa de Pequena Dimensão	122

9.2.4. Cenário 4 <i>Intranet</i> – Empresa de Grande Dimensão	123
10. CONCLUSÕES E TRABALHOS FUTUROS.....	125
REFERÊNCIAS BIBLIOGRÁFICAS	127
ANEXOS	129

ÍNDICE DE FIGURAS

Figura 1 - Esquema Geral do Sistema Atual	11
Figura 2 - <i>Software SCADA DSI - eneidaws</i> [aplicação proprietária]	14
Figura 3 - Exemplo do <i>HMI</i> de um sistema <i>SCADA</i> numa sala de controlo (GECCAI,2014) ..	20
Figura 4 - Pequeno exemplo de um sistema <i>SCADA</i> (Wikipédia, 2012)	21
Figura 5 - Exemplo de monitorização de sensores no software <i>WinCC</i> (Siemens AG, 2014) ..	24
Figura 6 - Editor de projetos do <i>Movicon X</i> (Exata,2010).....	25
Figura 7 - Exemplo de monitorização de sensores no <i>software ScadaBR</i> (ScadaBR, 2012).....	27
Figura 8 - Diagrama de Casos de Uso Aplicação Servidor	35
Figura 9 - Diagrama de Casos de Uso Aplicação Serviço de Alarmes	37
Figura 10 - Diagrama de Casos de Uso <i>Web Service</i>	41
Figura 11 - Diagrama de Casos de uso Ator Anónimo Aplicação <i>SCADA Web</i>	45
Figura 12 - Diagrama de Pacotes e Casos de Uso das Funcionalidades Gerais Aplicação <i>Web SCADA</i>	46
Figura 13 - Visão geral do sistema <i>e-Diagnostic software</i> para monitorização das redes de sensores.....	53
Figura 14 - Exemplo Estrutura de Dados baseada na arquitetura <i>Multi-Tenant Data</i>	55
Figura 15 - Diagrama arquitetura "N camadas"	56
Figura 16 - Exemplo do funcionamento da Arquitetura <i>Web API</i> (TDC2013, 2013).....	58
Figura 17 - Exemplo do Funcionamento da Arquitetura <i>MVC</i> (Feng Hua, 2014)	60
Figura 18 - Diagrama representativo da estrutura do módulo da aplicação servidor na arquitetura "N Camadas"	79
Figura 19 – Diagrama de atividades da aplicação servidor	81
Figura 20 - Estrutura física da aplicação servidor no <i>Visual Studio 2012</i>	82
Figura 21 - Diagrama de Classes do modelo mapeado pelo <i>Entity Framework</i>	84
Figura 22 - Aplicação Servidor	85
Figura 23 - Ícone da aplicação servidor na barra de tarefas	85
Figura 24 - Diagrama representativo da estrutura do módulo da aplicação serviço de alarmes na arquitetura "N Camadas"	86
Figura 25 – Diagrama de atividades da aplicação serviço de alarmes	87

Figura 26 - Estrutura física da aplicação Serviço de Alarmes no <i>Visual Studio 2012</i>	88
Figura 27 - Diagrama de Classes do modelo mapeado pelo <i>Entity Framework</i>	89
Figura 28 - Design da Aplicação Serviço de Alarmes.....	89
Figura 29 - Ícone da aplicação Serviço de alarmes na barra de tarefas.....	89
Figura 30 - Diagrama representativo da estrutura do Módulo <i>Web Service</i>	90
Figura 31 - Estrutura física da aplicação <i>Web Service</i> no <i>Visual Studio 2012</i>	91
Figura 32 - Diagrama de Classes do modelo mapeado pelo <i>Entity Framework</i>	93
Figura 33 - Resultado Pedido <i>GET</i> em <i>JSON</i>	93
Figura 34 - Diagrama representativo da estrutura do módulo da aplicação <i>SCADA Web</i>	94
Figura 35 - Diagrama representativo da estrutura de componentes principais do módulo da aplicação <i>SCADA Web</i>	95
Figura 36 - Estrutura física da aplicação <i>SCADA Web</i> no <i>Visual Studio 2012</i>	97
Figura 37 - Diagrama de Classes do modelo mapeado pelo <i>Entity Framework</i>	100
Figura 38 - Diagrama de Classes do modelo mapeado pelo <i>Entity Framework</i>	100
Figura 39 - Vista Principal da aplicação <i>SCADA Web</i>	101
Figura 40 - Esquema simplificado de um Posto de Transformação	105
Figura 41 - Posto de Transformação com tipologia de cabine usado no caso de estudo pela <i>eneidaws</i>	105
Figura 42 - Descrição dos sensores desenvolvidos para monitorização dos postos de transformação e suas características, <i>eneidaws</i> [documento interno da empresa]	107
Figura 43 - Descrição da <i>gateway</i> desenvolvida para fazer a ponte entre a rede de sensores e a aplicação servidor e suas características, <i>eneidaws</i> [documento interno da empresa].....	107
Figura 44 - Instalação dos sensores <i>EWS DTTI_reh</i> para medir a corrente em cada circuito do quadro geral do PT, <i>eneidaws</i> [documento interno].....	108
Figura 45 - Instalação do sensor <i>EWS DTH2O</i> para verificar a existência de inundação no PT, <i>eneidaws</i> [documento interno].....	109
Figura 46 - Instalação do sensor <i>EWS DTP</i> para controlar a abertura e fecho da porta no PT, <i>eneidaws</i> [documento interno].....	109
Figura 47 - Instalação do sensor <i>EWS DHT</i> para medir temperatura e humidade no PT, <i>eneidaws</i> [documento interno].....	109
Figura 48 - Instalação do sensor <i>EWS DTPD</i> para medir as descargas parciais no PT, <i>eneidaws</i> [documento interno]	109

Figura 49 - Visão geral da instalação dos sensores no PT, <i>eneidaws</i> [documento interno].....	110
Figura 50 - <i>Software SCADA DSI - eneidaws</i> [aplicação proprietária]	111
Figura 51 - Esquema representativo do cenário real da rede de sensores instalada no posto de transformação	112
Figura 52 - Exemplo Aplicação <i>SCADA Web</i> Página de <i>Login</i>	113
Figura 53 - Exemplo Aplicação <i>SCADA Web</i> Página Principal.....	114
Figura 54 - Exemplo de página de reporte onde se vê um gráfico dos dados de um sensor na rede do PT do caso de estudo	114
Figura 55 - Exemplo de página Aplicação <i>SCADA Web</i> Mapa Rede de Sensores	115
Figura 56 - Visualização dos últimos dados adquiridos na rede de sensor do PT do caso de estudo e estado de cada sensor	116
Figura 57 - Cenário 1 geral do sistema através da internet	121
Figura 58 - Cenário 2 geral do sistema através da internet, com aplicações servidores dedicadas às redes de sensores de cada cliente	122
Figura 59 - Cenário 3 geral do sistema instalado numa rede local <i>intranet</i> de um cliente.....	122
Figura 60 - Cenário 4 geral do sistema instalado numa rede local <i>intranet</i> de um cliente, com aplicações servidores dedicadas para cada uma das redes de sensores instaladas	123

ÍNDICE DE TABELAS

Tabela 1 - Calendarização de Tarefas.....	6
Tabela 2 - Grelha de Tarefas	7
Tabela 3 - Formato das Mensagens usado pela eneidaws	14
Tabela 4 - Requisitos aplicação servidor.....	33
Tabela 5 - Requisitos não funcionais aplicação servidor	36
Tabela 6 - Requisitos Aplicação Serviço de Alarmes	37
Tabela 7 - Requisitos não funcionais aplicação serviço de alarmes.....	38
Tabela 8 - Requisitos aplicação <i>Web Service</i>	39
Tabela 9 - Requisitos não funcionais aplicação <i>Web Service</i>	42
Tabela 10 - Requisitos Gerais Aplicação <i>SCADA Web</i>	43
Tabela 11 - Requisitos não funcionais aplicação <i>SCADA Web</i>	47
Tabela 12 - Relacionamento Empresa_Paises	70
Tabela 13 - Relacionamento Licenca_Empresa	70
Tabela 14 - Relacionamento UserProfile_Empresa.....	71
Tabela 15 - Relacionamento webpages_UserInRoles	71
Tabela 16 - Relacionamento Empresa_data_Rede	73
Tabela 17 - Relacionamento Rede_Localizacao	74
Tabela 18 - Relacionamento Sensor_Rede.....	74
Tabela 19 - Relacionamento Sensor_ModeloSensor	75
Tabela 20 - Relacionamento Sensor_Grupo.....	75
Tabela 21 - Relacionamento Entrada_Sensor.....	76
Tabela 22 - Relacionamento Entrada_LogMensagensServidor	76
Tabela 23 - Relacionamento Entrada_LogMensagensAlarmesServidor	77
Tabela 24 - Relacionamento Regras_Rede_Acao	77
Tabela 25 - Comparativo Aplicação <i>SCADA</i> Atual e Nova Aplicação <i>SCADA Web</i>	118
Tabela 26 - Comparativo Aplicação Servidor Atual e da Nova Aplicação Servidor	119

ABREVIATURAS

API – Application Programming Interface

ASCII – American Standard Code for Information Interchange

BD – Base de Dados

CAN – Controller Area Network

CLR – Common Language Runtime

CSS – Cascading Style Sheets

DCE – Data Communication Equipment

DEIS – Departamento de Engenharia Informática e de Sistemas

DLL – Dynamic-Link Library

DNP3 – Distributed Network Protocol, version 3

DNS – Domain Name Server

DS1 – Diagnostic Software 1

DTE – Data Terminal Equipment

ESTGOH – Escola Superior de Tecnologias e Gestão de Oliveira do Hospital

FCL – Framework Class Library

FDD – Feature Driven Development

GPRS – General Packet Radio Service

GPS – Global Positioning System

GSM – Global System for Mobile Communications

HMI – Human Machine Interface

HTML5 – HyperText Markup Language, version 5

HTTP – HyperText Transfer Protocol

IIS – Internet Information Services

IP – Internet Protocol

IPC – Instituto Politécnico de Coimbra

IPN – Instituto Pedro Nunes

ISEC – Instituto Superior de Engenharia de Coimbra

JSON – JavaScript Object Notation

LINQ – Language-**I**ntegrated **Q**uery
MCE – Mestrado em Comércio **E**letrônico
MVC – Model-**V**iew-**C**ontroller
ODBC – **O**pen **D**atabase **C**onnectivity
OPC – **O**bject **L**inking and **E**mbedding for **P**rocess **C**ontrol
ORM – **O**bject **R**elational **M**apping
PC – **P**ersonal **C**omputer
PDF – **P**ortable **D**ocument **F**ormat
PLC – **P**rogrammable **L**ogic **C**ontroller
PT – **P**ostos de **T**ransformação
SCADA – **S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition
SGBD – **S**istema de **G**estor de **B**ase de **D**ados
SMS – **S**hort **M**essage **S**ervice
SOAP – **S**imple **O**bject **A**ccess **P**rotocol
SQL – **S**tructured **Q**uery **L**anguage
TCP – **T**ransmission **C**ontrol **P**rotocol
UDP – **U**ser **D**atagram **P**rotocol
UML – **U**nified **M**odeling **L**anguage
VBA – **V**isual **B**asic for **A**pplications
XML – **e**Xtensible **M**arkup **L**anguage

1. INTRODUÇÃO

1.1. ENQUADRAMENTO

O presente relatório, do projeto “*e-Diagnostic Software* - aplicação *web* para monitorização de sensores em unidades industriais”, referente às disciplinas de “Projeto ou Estágio I e II”, pertencentes aos primeiro e segundo semestres, insere-se no Mestrado em Comércio Eletrónico. Este mestrado é realizado em parceria com o Instituto Superior de Engenharia de Coimbra e a Escola Superior de Tecnologias e Gestão de Oliveira do Hospital, ambas unidades orgânicas do Instituto Politécnico de Coimbra, sendo este mestrado lecionado no Departamento de Engenharia Informática e de Sistemas (DEIS)³, do Instituto Superior de Engenharia de Coimbra.

O projeto “*e-Diagnostic Software* - aplicação *web* para monitorização de sensores em unidades industriais” surge através da parceria com a empresa *eneidaws wireless & sensors*⁴, sendo esta uma *startup* presente no mercado desde do ano 2012, sediada nas instalações do Instituto Pedro Nunes⁵ (IPN) - Associação para a Inovação e Desenvolvimento em Ciência e Tecnologia.

A área de negócio da *eneidaws* é engenharia de instrumentação e comunicação industrial com uma divisão especializada no desenvolvimento de *hardware* e uma divisão especializada no desenvolvimento de *software* para monitorização de equipamentos e estruturas. Focada no setor industrial, a *eneidaws* tem uma experiência diversificada em setores (óleo e gás, indústria química, energia, pasta e papel, minas, águas e resíduos), com inúmeras prestações desde a eletricidade e eletrónica industrial, automação e instrumentação à comunicação em tempo real através de infraestruturas *wireless* fiáveis e robustas.

Na divisão especializada no desenvolvimento de *hardware*, a *eneidaws* tem três categorias de dispositivos: sensores, unidades de controlo e *gateways*. Com base nestas três categorias de *hardware*, a *eneidaws* projeta e instala sistemas, que permitem, de uma forma simples e económica, recolher informação sobre os processos produtivos de modo contínuo, em tempo real, e possibilitam o acesso à informação em qualquer local. Estes sistemas constituem soluções nomeadamente para as áreas de manutenção preditiva, eficiência energética, ambiente/controlo de emissões, segurança, controlo de acessos e monitorização da integridade estrutural.

³ Departamento de Engenharia Informática e de Sistemas. Disponível no endereço: <http://deis.isec.pt/>

⁴ Neste relatório são citadas empresas e produtos que são marcas registadas. Dada a sua quantidade, considera-se desnecessário estar a identificar cada um, sendo citada esta informação nesta nota de rodapé.

⁵ Instituto Pedro Nunes. Disponível no endereço: <https://www.ipn.pt/>

Na divisão especializada no desenvolvimento de *software*, a *eneidaws* produz o seu próprio *software* capaz de analisar e monitorizar os dados adquiridos pelas redes de sensores instaladas nas unidades industriais. Neste contexto, a *eneidaws* conta atualmente com uma aplicação servidor responsável por receber os dados provenientes das redes de sensores e armazenar essa informação em base de dados. Para consulta da informação armazenada, a *eneidaws* tem o *software Supervisory Control And Data Acquisition (SCADA)*, denominado *Diagnostic Software 1 (DS1)*, desenvolvido em tecnologias *Windows*, que permite unicamente o funcionamento em modo local em sistemas operativos *Windows*. Este *software* tem como finalidade a visualização de todos os dados adquiridos pelas redes de sensores dos diferentes sistemas instalados nas unidades industriais, possibilitando posteriormente a análise e tratamento desses dados.

Atualmente, a aplicação *SCADA DS1* carece de funcionalidades e opções que permitam aos utilizadores ter mais segmentação na informação adquirida pelas redes de sensores; por outro lado, esta aplicação necessita de mais funcionalidades e opções que correspondam às necessidades dos clientes e da própria empresa.

Neste contexto, tendo em conta as lacunas e necessidades existentes na aplicação *SCADA DS1*, verificou-se a oportunidade de o aluno, sendo funcionário da *eneidaws* e finalista do MCE, propor o desenvolvimento do projeto “*e-Diagnostic Software - aplicação web* para monitorização de sensores em unidades industriais”. Este projeto tem como objetivo otimizar o sistema atual da *eneidaws* para monitorização de sensores instalados nas unidades industriais, para ser mais flexível e contemplar mais funcionalidades e opções.

Para isso, propôs à *eneidaws* o desenvolvimento principal de uma nova aplicação *SCADA* numa vertente *web*, multiplataforma, com maior leque de funcionalidades e opções que venham a corresponder às necessidades dos clientes. Propôs-se ainda o desenvolvimento de uma nova aplicação servidor capaz de receber e adquirir dados das redes de sensores, com mais funcionalidades e o melhoramento geral de todo o sistema.

A *eneidaws* propôs a integração, neste projeto, do caso de estudo que tem atualmente, o qual tem como finalidade a monitorização dos postos de transformação energéticos de um cliente seu, a nível ambiental, energético, de segurança, de qualidade do serviço, entre outros. Com a integração deste caso de estudo neste projeto, torna-se possível validar com informação real se os objetivos pretendidos foram atingidos.

A existência da possibilidade de integrar este projeto na tese final do aluno permitiu também a aproximação de entidades externas do tecido empresarial de Coimbra à instituição de ensino.

1.2. ENTIDADES ENVOLVIDAS

1.2.1. ISEC

O ISEC é uma instituição de ensino superior que tem como missão e valores a transmissão e difusão de conhecimento na formação referente ao domínio das engenharias, como se pode ver na informação original obtida, no *site* do ISEC.

“O ISEC tem como missão a criação, transmissão e difusão de cultura, ciência e tecnologia, cabendo-lhe ministrar uma formação de nível superior para o exercício de atividades profissionais no domínio da Engenharia e promover o desenvolvimento da região em que se insere.

Como visão institucional, procura ser uma referência de excelência no ensino, reconhecido nacional e internacionalmente por serviços de qualidade e relevância social, com práticas flexíveis, criativas e inovadoras. Pretende ainda ser um parceiro privilegiado das organizações empresariais e das famílias da região onde se insere pela orientação eminentemente prática, fundada num rigoroso conhecimento teórico, que imprime a todas as suas atividades.

Os valores fundamentais pelos quais se rege o ISEC são a cidadania, a qualidade, a busca constante da valorização, motivação e atualização pedagógica, científica e tecnológica dos seus recursos, o bom relacionamento e a disponibilidade para com os estudantes e as organizações suas parceiras e a preocupação com o desenvolvimento social e económico da região onde está inserido.”⁶

1.2.2. ESTGOH

A ESTGOH é uma instituição de ensino superior que tem como missão e valores a aposta na formação de excelência, como se pode ver na informação original obtida, no *site* da ESTGOH.

“A ESTGOH é uma unidade orgânica do IPC e tem por missão a formação de quadros altamente qualificados, por intermédio da lecionação de cursos de especialização tecnológica, licenciaturas e formação pós-graduada.

A aposta na inovação dos métodos e materiais de trabalho, a contínua formação dos seus docentes e a adaptação dos planos curriculares dos cursos às exigências do mercado de

⁶ Missão e Valores do Instituto Superior Engenharia de Coimbra. Disponível no endereço: <http://www.isec.pt/isec/>

trabalho cada vez mais competitivo, são as bases da estratégia da instituição, alcançando desta forma a completa satisfação da sua comunidade académica.

A ESTGOH promove uma constante adaptação da sua oferta educativa às necessidades e expectativas dos seus alunos, da região e do país, apostando na preparação adequada dos seus formandos, a quem garante uma elevada taxa de empregabilidade na conclusão dos cursos.”⁷

1.2.3. *eneida wireless & sensors*

A *eneidaws* é uma *startup*, no mercado desde do ano 2012, sediada nas instalações do IPN. Tem como mercado alvo as empresas do setor industrial, sejam estas do mercado nacional ou internacional. Neste último, a *eneidaws* tem feito uma grande aposta na angariação de novos clientes para exportação dos seus produtos e serviços. Para isso a *eneidaws* tem apostado fortemente na sua componente comercial direcionada a outros mercados, apresentando novo *site*, com idioma em inglês, onde são expostos os seus produtos e serviços.

Com vista a posicionar-se de uma forma credível perante os atuais e futuros clientes, a *eneidaws* tem reunido uma série de certificações, no sistema de gestão da qualidade e segurança estabelecido pela *ISO9001:2008* e *OHSAS 18001:2007*, certificados que garantem o cumprimento e melhoria contínua de seus produtos e serviços, satisfação do cliente, segurança e saúde dos seus funcionários. Tem ainda alguns produtos certificados de acordo com a diretiva *ATEX 94/9 / CE* e *IECEX* inserida no sistema de qualidade e gestão de segurança *ATEX ISO / IEC80079-34: 2011*, para definição de algumas regras e exigências no desenvolvimento dos seus produtos.

Sendo a *eneidaws* especialista na área de engenharia de instrumentação e comunicação industrial, conta com uma divisão especializada no desenvolvimento de *hardware* (apresentando atualmente uma série de produtos produzidos por si em carteira)⁸ e uma divisão especializada no desenvolvimento de *software* para monitorização das redes de sensores instaladas nas unidades industriais.

⁷ Missão e Valores da Escola Superior de Tecnologia e Gestão de Oliveira do Hospital. Disponível no endereço: <http://websrv2.estgoh.ipc.pt/portal2/escola/miss-o-vis-o-e-pol-tica-da-qualidade/>

⁸ *eneida wireless & sensors*. Informação sobre os produtos, disponível no endereço: <http://eneidaws.com/portfolio/>

Na divisão especializada no desenvolvimento de *hardware*, a *eneidaws* apresenta-se com três categorias de dispositivos: sensores, unidades de controlo e *gateways*. Os sensores são dispositivos responsáveis pela recolha de dados sobre os processos produtivos. As unidades de controlo são dispositivos que têm controlador lógico programável, que possibilita a programação destes dispositivos, permitindo que estes tenham controlo de diferentes níveis de complexidade e desempenhem determinadas funções e ações. As *gateways* são dispositivos que têm como função fazer o roteamento de mensagens dentro da rede de sensores e para fora da rede de sensores bidireccionalmente. Com base nestas três categorias de *hardware*, a *eneidaws* projeta e instala sistemas, que permitem, de uma forma simples e económica, recolher informação sobre os processos produtivos de modo contínuo, em tempo real.

Na divisão especializada no desenvolvimento de *software*, a *eneidaws* produz o seu próprio *software* capaz de analisar e monitorizar os dados adquiridos pelas redes de sensores instaladas nas unidades industriais. Neste contexto, a *eneidaws* tem atualmente uma aplicação servidor para receber os dados das redes de sensores e uma aplicação *SCADA DSI*, para supervisionar e monitorizar a informação adquirida das redes de sensores.

A *eneidaws*, para além dos produtos e soluções que tem ao seu dispor desenvolvidas por si, produz *hardware* e *software* à medida das necessidades específicas dos seus clientes. Esta produção de *hardware* e *software* verifica-se em projetos que tenham necessidades específicas e quando nenhum dos produtos e soluções que a *eneidaws* tem se mostra adequado às necessidades do cliente. Neste caso, são desenvolvidos protótipos para esses clientes, nos quais são realizados todos os testes necessários, para garantir a satisfação do cliente. Sempre que possível, a *eneidaws* tenta converter estes protótipos em novos produtos, para que mais tarde possam ser usados em novas soluções por outros clientes com necessidades semelhantes. Neste sentido será uma grande vantagem para a *eneidaws* ter *hardware* e *software* suficientemente portátil e flexível para que estes possam ser usados num maior leque de soluções.

1.3. OBJETIVOS

A elaboração deste projeto tem como objetivo o desenvolvimento de um novo sistema *SCADA* para supervisão, monitorização e controlo das redes de sensores implementadas pela *eneidaws* e instaladas nas unidades industriais. Pretende-se que este novo sistema apresente novas funcionalidades relativamente ao sistema atual da *eneidaws*, que seja acessível em qualquer local via *web* e que seja compatível com o *hardware* atual que a empresa possui. As aplicações a ser desenvolvidas para este novo sistema devem ter uma arquitetura modular, apresentando mais opções que permitam uma melhor análise no tratamento dos dados referentes às redes de sensores. Este novo sistema deve apresentar uma estrutura flexível, segura, com informação centralizada.

Para se atingir o objetivo da elaboração do novo sistema para supervisão, monitorização e diagnóstico das redes de sensores, será necessário desenvolver especificamente as seguintes aplicações:

- Uma aplicação *SCADA Web*, responsável pela supervisão, controlo e visualização do histórico de dados gerados pelas diferentes redes de sensores;
- Uma aplicação servidor, capaz de receber todas as mensagens proveniente das redes de sensores;
- Uma aplicação serviço de alarmes, responsável por tratar alarmes⁹;
- Uma aplicação *Web Service*, responsável por disponibilizar informação de todas as redes de sensores através de pedidos por *Hypertext Transfer Protocol (HTTP)*;
- Um novo modelo de dados que expresse a lógica de negócio de todo o sistema;

1.4. METODOLOGIA E PLANEAMENTO DE TAREFAS

No desenvolvimento deste projeto, não foi utilizada uma metodologia de desenvolvimento de *software*, como por exemplo o *SCRUM* (Wikipédia,2011) ou *Feature Driven Development (FDD)* (Agile Modelling,2014) durante a elaboração do projeto, dado este não ter sido realizado por uma equipa de desenvolvimento. No entanto, foi realizado um plano de trabalhos entre o aluno e o orientador com tarefas a cumprir, e algumas reuniões do aluno e do orientador com empresa.

1.4.1. Calendarização de Tarefas

Tabela 1 - Calendarização de Tarefas

Tarefas	Meses													
	N	N+1	N+2	N+3	N+4	N+5	N+6	N+7	N+8					
T1														
T2														
T3														
T4														
T5														
T6														
T7														
Metas	M1	M2	M3	M4									M5	M6

⁹ O exemplo de um alarme, poderá ser gerado por um valor de temperatura lido por um sensor estar acima do valor que foi definido como normal, originando um alarme.

Tabela 2 - Grelha de Tarefas

Data início do projeto 01/02/2014 e data fim em 30/10/2014		
INI		Início dos Trabalhos
M1	(INI + 9 Semana)	Tarefa T1 - Análise de Requisitos terminada
M2	(INI + 4 Semana)	Tarefa T2 – Análise das Ferramentas e Tecnologias terminada
M3	(INI + 2 Semana)	Tarefa T3 - Base de Dados terminada
M4	(INI + 4 Semana)	Tarefa T4 - Conceção dos <i>interfaces</i> gráficos terminada
M5	(INI + 14 Semana)	Tarefa T5 - Desenvolvimento e Codificação das Aplicações terminada
M6	(INI + 6 Semana)	Tarefa T6 - Testes terminada
M7	(INI + 4 Semana)	Tarefa T7 - Relatório terminada

1.4.2. Descrição das Tarefas

T1 – Análise de Requisitos

- T1.1 – Análise e estudo dos dispositivos da *eneidaws* (sensores, unidades de aquisição e *gateways*).
- T1.2 – Análise da estrutura das mensagens e mecanismo de comunicação usado pela empresa para as diferentes redes de sensores reportarem os seus dados para a aplicação servidor e posterior armazenamento dos dados em base de dados.
- T1.3 – Análise do *software* existente atualmente na empresa, explorando as melhorias que poderiam ser implementadas e quais as funcionalidades que é necessário serem migradas, relativa à aplicação servidor e à aplicação *SCADA DSI* para consulta dos dados armazenados pelo servidor.
- T1.4 – Estado da arte, pesquisa sobre sistemas *SCADA* atualmente existentes e suas funcionalidades.
- T1.5 – Levantamento de todos os requisitos referentes à aplicação servidor responsável por receber e comunicar informação com as redes de sensores.

- T1.6 – Levantamento de todos os requisitos referentes à aplicação serviço de alarmes e funcionamento esperado.
- T1.7 – Levantamento de todos os requisitos referentes ao *Web Service* e funcionamento esperado.
- T1.8 – Levantamento de todos os requisitos referentes às funcionalidades gerais da aplicação *SCADA Web* e todos os requisitos específicos para integração do caso de estudo que a empresa tem presentemente.
- T1.9 – Compilação de todos os dados recolhidos; análise do problema proposto; especificação e estudo dos requisitos levantados; prototipagem das aplicações e esboço da arquitetura a implementar; definição de funcionalidades/tarefas para desenvolvimento das quatro aplicações.

T2 – Análise das Ferramentas e Tecnologias - Escolha das ferramentas e tecnologias a utilizar para o desenvolvimento do projeto. Instalação, aprendizagem e realização de alguns tutoriais para compreender o funcionamento de ferramentas e tecnologias a utilizar.

T3 – Base de Dados - Modelação e desenho da base de dados, tendo em conta os requisitos e a arquitetura das aplicações, para armazenamento de toda a informação.

T4 – Conceção dos *interfaces* gráficos - Conceção dos diferentes *interfaces* gráficos para a implementação das diferentes aplicações.

T5 – Desenvolvimento e Codificação das Aplicações

- T5.1 – Desenvolvimento e codificação da aplicação servidor, para receber e comunicar com a rede de sensores.
- T5.2 – Desenvolvimento e codificação da aplicação Serviço de Alarmes, para o utilizador ser notificado quando existirem alarmes numa rede.
- T5.3 – Desenvolvimento e codificação do *Web Service*, para disponibilizar informação sobre a rede de sensores através de pedidos *HTTP* por *GET* e *POST*.
- T5.4 – Desenvolvimento e codificação da aplicação *SCADA Web*, para consulta de informação e configuração das redes de sensores.

T6 – Testes - Testes funcionais gerais realizados a todo o sistema para garantir a robustez do mesmo. Utilização da rede atualmente em funcionamento (caso de estudo da empresa, capítulo 8), para garantir que todo o sistema está implementado com base nos requisitos estabelecidos. Os testes foram sendo realizados ao mesmo tempo que cada funcionalidade e opção foram finalizadas.

T7 – Relatório - Elaboração do relatório final do projeto. A produção do relatório foi sendo realizada ao longo do desenvolvimento de todo o sistema.

1.5. ESTRUTURA DO RELATÓRIO

No capítulo 1, referente à introdução, é feito o enquadramento do projeto, entidades envolvidas, objetivos a atingir, planeamento dos trabalhos e estrutura do relatório.

No capítulo 2 é feita a avaliação do sistema atualmente presente na empresa, em particular a aplicação servidor e a aplicação *SCADA DSI*, a perceção de limitações existentes e possíveis melhorias nas mesmas. São abordados ainda os requisitos iniciais pretendidos pela empresa na implementação do novo projeto para monitorização de sensores instalados nas unidades industriais.

No capítulo 3 analisa-se o que são sistemas *SCADA*, sendo dados alguns exemplos de sistemas deste género existentes no mercado e feita uma análise comparativa destes sistemas. Apresenta-se ainda uma breve comparação do sistema *SCADA* atual da *eneidaws* com as características principais dos sistemas apresentados.

No capítulo 4 é descrita a solução proposta para o desenvolvimento do novo sistema, com as vantagens esperadas pela empresa, com base nos requisitos iniciais vistos no capítulo 2. É apresentada a proposta dos requisitos e protótipos das aplicações para implementação do novo sistema.

No capítulo 5 é dada uma descrição da arquitetura onde irá assentar o novo sistema, atendendo aos requisitos recolhidos e analisados no capítulo 4. É explicada a utilização de determinadas tecnologias e ferramentas usadas ao longo deste projeto.

No capítulo 6 é apresentado o modelo de base de dados, desenhado para este projeto e onde será mostrada a estrutura do modelo e respetivo motor de base de dados a utilizar.

No capítulo 7 é concebida uma explicação para cada um dos módulos do sistema desenvolvidos que constituem o projeto.

No capítulo 8 é explicado o caso de estudo (postos de transformação) existente na empresa e feita a integração deste caso de estudo neste projeto.

No capítulo 9 é realizada a análise crítica do trabalho elaborado para o presente projeto e apresentados possíveis cenários de aplicação deste projeto demonstrando a versatilidade deste novo sistema.

No capítulo 10 é descrita a conclusão de todo o projeto desenvolvido e descritos os possíveis trabalhos futuros.

Na secção referências bibliográficas é elaborada a listagem das referências usadas no desenvolvimento deste relatório.

Na secção anexos são apresentados cinco anexos, com informação complementar à estrutura do relatório principal.

- Anexo A - Análise de Requisitos, informação complementar da análise de requisitos com diagramas de caso de uso.
- Anexo B - Prototipagem das Aplicações, exemplo dos protótipos definidos para cada aplicação.
- Anexo C - Modelo de Dados, diagramas sobre o modelo conceptual e modelo físico.
- Anexo D - Manual de utilizador definido para cada uma das aplicações.
- Anexo E - Testes, realizados a todas as aplicações desenvolvidas e respetiva análise dos testes realizados.

2. CONTEXTO E DOMÍNIO DA APLICAÇÃO

Neste capítulo apresenta-se o funcionamento do sistema atual da *eneidaws*. Aqui serão identificadas as principais limitações do sistema atual e apresentadas as perspectivas de melhorias a realizar no sistema.

Demonstrar-se-á ainda quais as principais necessidades evidenciadas pela *eneidaws*, referentes à elaboração deste projeto.

2.1. DESCRIÇÃO DO SISTEMA ATUAL

A *eneidaws*, enquanto empresa de engenharia e instrumentação, produz várias gamas de *hardware* (sensores, unidades de controlo, *gateways*), posteriormente instaladas nas unidades industriais, para realizarem a aquisição dos dados provenientes dos locais onde estão alojados. Para análise dos dados, a *eneidaws* tem um *software SCADA DS1* complementar para tratamento e monitorização desses dados.

Para facilitar a compreensão do sistema atual, dividiu-se o mesmo em três partes, com uma descrição sucinta de cada uma das partes vistas na figura 1.

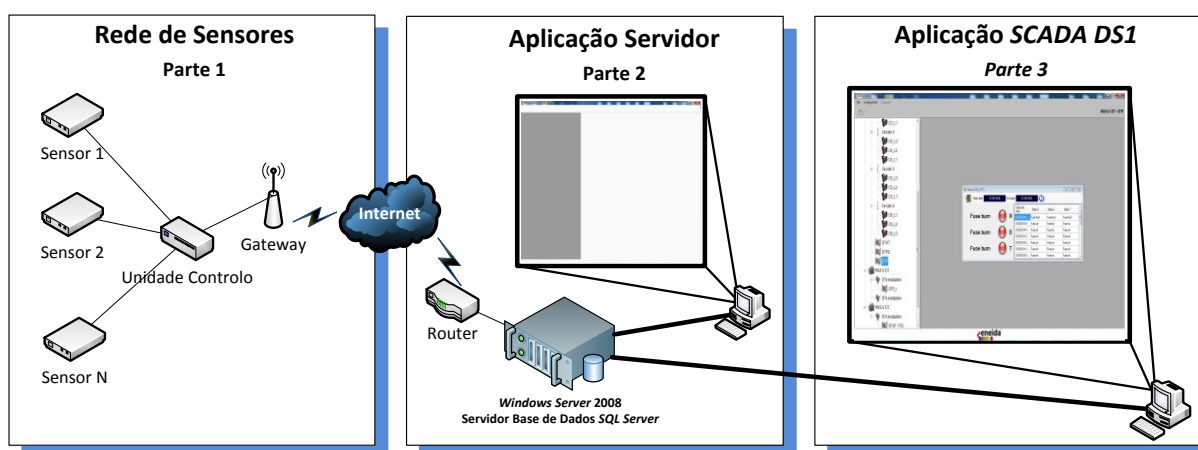


Figura 1 - Esquema Geral do Sistema Atual

Parte 1 - Rede de Sensores

A rede de sensores vista na figura 1 representa, de uma forma simplificada, o *hardware* instalado nas unidades industriais pela *eneidaws*. O *hardware* desenvolvido encontra-se subdividido em três tipos: sensores, unidades de controlo e *gateways*.

A comunicação entre as redes de sensores e a aplicação servidor vista na figura 1 é feita através de ligações *socket*¹⁰ por *TPC/IP*¹¹. Cada rede de sensores é cliente da aplicação servidor, onde as mesmas se conectam a um determinado *Internet Protocol (IP)* e porto de comunicação. Assim que a conexão seja estabelecida, é feita comunicação de dados entre a rede de sensores e o servidor, de forma bidirecional.

A configuração típica de uma rede de sensores vista na figura 1, implementada pela *eneidaws*, tem o seguinte *hardware* na sua estrutura:

- **Sensor** - Os sensores têm como responsabilidade adquirir dados dos locais onde estão instalados; por exemplo, pressão, temperatura, corrente, tensão, humidade, entre outros. Posteriormente, estes dados são submetidos para as unidades de controlo;
- **Unidade de Controlo** - As unidades de controlo, com base nos dados adquiridos pelos sensores, desempenham duas funções: mediante o algoritmo com que foram programados, podem ter determinadas funções e ações para o contexto em que estão inseridos, por exemplo ligar ou desligar o sistema; por outro lado, estas unidades têm de comunicar todas as informações recebidas pelos sensores e encaminhar esses dados para o equipamento *master* da rede, que é a *gateway*;
- **Gateway** - As *gateways* têm como função estabelecer uma conexão¹² por *socket* com a aplicação servidor e fazer a ponte das redes de sensores entre o *hardware* e o *software*, e vice-versa. Estas *gateways* têm como função enviar todos os dados provenientes das redes de sensores e submeter os dados à aplicação servidor. Devem igualmente ser capazes de receber informações da aplicação servidor e comunicar essa informação a determinada unidade de controlo (por exemplo, o *software* envia instrução para desligar todo o sistema).

¹⁰ Um *socket* pode ser usado em ligações de redes de computadores ou outros equipamentos para estabelecer um elo bidirecional de comunicação entre dois programas ou dispositivos que utilizam o mesmo protocolo de comunicação estando ligados na mesma rede.

¹¹ **TCP/IP** (também chamado pilha de protocolos TCP/IP) é um conjunto de protocolos de comunicação muito usado nos dias de hoje entre rede de computadores e outros equipamentos.

¹² A conexão entre a *gateway* e a aplicação servidor pode ser feita de várias formas, dependendo do meio físico onde está inserida a rede. Por exemplo, pode ser feita por ligação; *Global System for Mobile Communications (GSM)* via *General Packet Radio Service (GPRS)*; por rede cablada *Ethernet*; *Wireless*, entre outros; dependerá sempre do contexto do meio físico.

Relativamente à estrutura física de comunicação dentro de uma rede de sensores, existem diferentes meios físicos e protocolos de comunicação que podem ser usados. No caso, a *eneidaws* utiliza, na maioria das suas redes, a comunicação por meio físico rede *Wireless*, sendo este sistema empregue, na maioria dos casos, com recurso ao protocolo de comunicação *TCP/IP*, por ser mais fácil e económica a instalação, garantindo do mesmo modo a fiabilidade na aquisição dos dados. No entanto, outros meios físicos e protocolos de comunicação podem ser usados dentro de uma rede, dependerá sempre do contexto e do meio físico para a qual se esteja a projetar uma rede.

Parte 2 - Aplicação Servidor

A aplicação servidor vista na figura 1, desenvolvida em *Windows Forms*, tem como função receber os dados provenientes da rede de sensores através de ligações estabelecidas *TCP/IP* por *socket*. Os dados recebidos são armazenados no motor de base de dados *Microsoft SQL Server* (Microsoft, 2014).

Os dados guardados na base de dados servem para posterior consulta na aplicação *SCADA DSI*. A aplicação servidor é configurada com o *IP* e porto para que as redes sensores clientes possam conectar-se a si e estas possam enviar os dados adquiridos. O servidor tem ainda a responsabilidade de submeter informação para uma determinada rede, caso seja solicitado alguma informação ou ação (por exemplo, desligar alarme, pedir valores de um determinado sensor, etc.).

O formato das mensagens (tabela 3) trocadas entre a rede de sensores e a aplicação servidor está em hexadecimal. Para facilitar o processo de comunicação, a *eneidaws* criou a sua própria estrutura de mensagens para “interpretação” e estruturação das mensagens de dados internos na rede física ou para comunicação da rede de sensores com o servidor, e vice-versa. Após tratamento das mensagens de dados, os atributos principais têm a seguinte apresentação, como se pode ver:

- ID Rede – ID da Rede representa de onde provém a mensagem (todas as redes ligadas ao servidor têm um ID único);
- ID Sensor – ID único do Sensor que enviou a mensagem;
- Código de Função – Representa a entrada do sensor com ID único que está a adquirir dados (por exemplo, código função 231 representa a entrada analógica 1 do sensor ID 10);
- Valor Adquirido – Representa o valor que foi recolhido na entrada de um determinado sensor.
- Valor Adquirido – Representa o valor que foi recolhido na entrada de um determinado sensor.

Tabela 3 - Formato das Mensagens usado pela eneidaws

Formato da Mensagem			
1	2	3	4
ID Rede	ID Sensor	Código Função	Valor Adquirido

Parte 3 - Aplicação SCADA DSI

Na aplicação *SCADA DSI* desenvolvida em *Windows Forms* vista na figura 2, referente ao atual sistema da *eneidaws*, permite configurar as diferentes redes de sensores, consultar e analisar os dados obtidos nas redes de sensores. A interligação desta aplicação com o resto do sistema é feita pelo motor de base de dados, destino de dados onde são depositados todos os dados adquiridos pelo servidor através da rede de sensores, como é demonstrado de forma simplificada na figura 1.

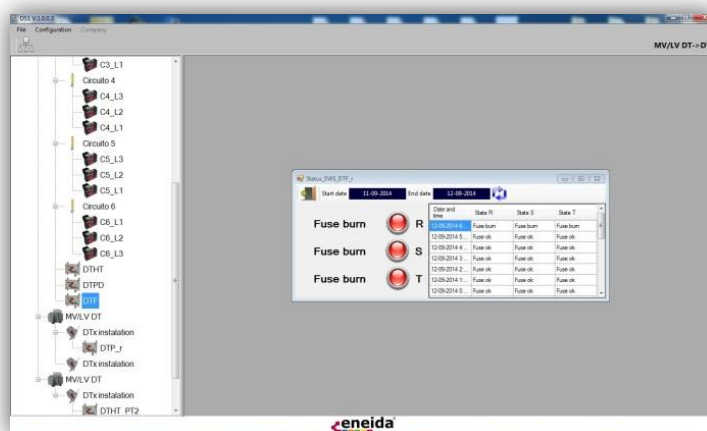


Figura 2 - Software SCADA DSI - eneidaws [aplicação proprietária]

2.2. LIMITAÇÕES DAS APLICAÇÕES NO SISTEMA ATUAL

2.2.1. Limitações da Aplicação SCADA DSI

A aplicação *DSI* permite configurar redes de sensores e visualizar o histórico de dados adquiridos das redes. No entanto, apresenta as seguintes limitações:

- Inexistência de controlo de acessos: qualquer utilizador pode abrir a aplicação e efetuar as configurações que pretenda, seja administrador da empresa ou um funcionário. Não existindo a possibilidade de ter o registo de atividades do utilizador dentro do sistema, este facto representa, assim, uma falha de segurança;
- Inexistência de áreas bem definidas para administração da aplicação e áreas para consulta de informação, mediante o nível de privilégios do utilizador;
- O facto da aplicação *SCADA DSI* ser desenvolvida em tecnologia *Windows Forms* é uma grande limitação, porque esta aplicação só funcionará em modo local em sistemas operativos *Windows*, não funcionando em outros sistemas operativos.
- Inexistência de uma aplicação com tecnologias *Web*, que permita que a mesma seja multiplataforma e multidispositivo; com informação centralizada; onde seja disponibilizada a aplicação *SCADA* via *Web* de forma fácil aos utilizadores. Sendo aplicação *Web* com a informação centralizada, torna-se mais fácil o processo de correções e atualizações para todos os utilizadores;
- Cada utilizador que tenha de usar a aplicação *SCADA DSI* é obrigado a instalar esta aplicação localmente. Tornar-se-á um problema a longo prazo porque sempre que exista uma nova correção ou atualização à aplicação *DSI*, será sempre necessário efetuá-la computador a computador, ou seja, quanto maior for o número de utilizadores maior será o tempo despendido;
- A aplicação *DSI*, quando precisa de aceder a dados das redes de sensores armazenados num servidor de base de dados remoto, produz ocasionalmente erros e a performance para tratamento dos dados não é boa;
- A visualização dos gráficos obtidos a partir do histórico de dados dos sensores é lenta, mesmo que a aplicação esteja instalada localmente onde se encontra o servidor de base de dados;
- Inexistência de um *dashboard* onde possam ser visualizados os últimos dados de uma determinada rede de sensores, com noção do estado dos sensores (por exemplo, se estão em alarme ou não);
- Falta de automatismo para importação e exportação de redes de sensores: cada rede que for configurada terá de o ser manualmente;
- Inexistência de uma área para criar regras para alarmes entre sensores de uma mesma rede ou entre sensores de uma outra rede.
- Inexistência de uma área que permita gerir e visualizar alarmes ocorridos numa rede;

- A aplicação não permite exportar os dados recolhidos dos sensores, para tratamento e análise destes dados;
- Inexistência de uma área de reporte para efetuar consultas específicas de determinada informação;
- O desenvolvimento de novas funcionalidades e opções é demorada, dado a estrutura da aplicação *DSI* não apresentar uma arquitetura modular e flexível;
- Inexistência de um *Web Service*, por um lado para acesso e consulta de informação relativa às redes de sensores e por outro para facilitar a integração desta informação com outros sistemas.

2.2.2. Limitações da Aplicação Servidor

Atualmente, no sistema *SCADA* da *eneidaws*, na ocorrência de um alarme numa rede de sensores, a única forma de o utilizador ser notificado com um alarme será exclusivamente por *Short Message Service (SMS)* enviado através do equipamento *master (gateway)* da rede de sensores, não existindo nenhuma outra forma no sistema capaz de notificar o utilizador.

Deste modo, considera-se problemático que só a rede de sensores consiga notificar o utilizador. Na eventualidade de o serviço de envio de *SMS* usado pelo equipamento *master* da rede falhar, o utilizador nunca será notificado da existência de alarmes. Neste contexto, apesar de a aplicação servidor atual funcionar convenientemente na troca de mensagens bidireccionalmente entre a rede de sensores e a parte aplicacional e armazenamento dos dados, é considerado uma limitação o servidor não ter capacidade de analisar alarmes e notificar por *e-mail* ou *SMS* o utilizador. Existe, desta forma, a possibilidade de capacitar a aplicação servidor para notificar o utilizador por *e-mail* e *SMS* na ocorrência de alarmes, conseguindo-se garantir com maior probabilidade que o utilizador será notificado nestas circunstâncias.

Desta forma consegue-se criar redundância no sistema de notificação de alarmes, ou seja, caso a rede sensores falhe no envio do alarme, o servidor envia o alarme, ou ambos enviam o alarme. Uma situação que pode ocorrer é ambos não enviarem alarmes; neste caso, a probabilidade de esta situação acontecer será mais baixa, porque temos partes diferentes do sistema capazes de analisar e enviar alarmes.

Na eventualidade de a aplicação servidor não ter capacidade de desempenhar a análise de ocorrência de alarmes, uma opção será criar uma aplicação exclusiva para verificar a ocorrência de alarmes, com notificação por *e-mail* ou *SMS* ao utilizador. Em suma, é importante ter em conta que se está a trabalhar com sistemas críticos, pelo que a existência de redundância no sistema de notificação de alarmes é muito importante.

2.3. REQUISITOS INICIAIS DA EMPRESA

O levantamento de todos os requisitos inicialmente pretendidos pela empresa foi realizado através de entrevistas na empresa, por observação direta e *brainstorming*. Pretendeu-se desta forma, numa fase inicial, compreender as necessidades chave evidenciadas pela empresa e daí partir para uma análise mais profunda e decisiva, com vista a uma solução concreta.

A lista de requisitos inicialmente pretendidos pela empresa é a seguinte:

- A nova aplicação *SCADA Web* deve ter as seguintes funcionalidades gerais: gestão de utilizadores; gestão de empresas/clientes; gestão de licenças; gestão de alarmes; consulta de histórico de alarmes e histórico de dados dos sensores de uma rede em gráficos e tabelas; gestão de redes e mapa com georreferenciação das redes de sensores instaladas;
- Deve haver um módulo *Web* para definir permissões de acesso à aplicação: os utilizadores devem ter vários níveis de privilégios; o utilizador administrador tem privilégios elevados, tendo controlo sobre a aplicação, enquanto o utilizador normal tem privilégios baixos, podendo apenas consultar informação;
- Deve ser feito o desenvolvimento de uma nova aplicação servidor, mantendo a estrutura de mensagens como está atualmente, e a comunicação deve ser feita por *socket* entre a rede de sensores e o *servidor* (como explicado na secção 2.1). Desta forma garante-se que as novas aplicações venham a funcionar com o atual e novo *hardware* desenvolvido pela *eneidaws*;
- O desenvolvimento das aplicações para o novo sistema deve ter como base o recurso a tecnologias e ferramentas para ambientes *Windows*. Esta situação verifica-se porque a empresa já fez um investimento em ferramentas de desenvolvimento como *Visual Studio* e *software* de base de dados *SQL Server*. Por esta razão, a empresa não vê vantagem no desenvolvimento em outras tecnologias sem ser *Windows*;
- O sistema a implementar deve ser capaz de funcionar com vários clientes, com recurso a um *interface Web*.
- Deve existir acesso à informação guardada no servidor de base de dados, referente à rede de sensores através de *Web Service*, para consulta e integração com outros sistemas;
- Os dados recolhidos das diversas redes de sensores dos vários clientes devem ser guardados de forma independente (cada cliente terá a sua fonte de dados). Deve

existir uma outra base de dados comum a todos os clientes, onde estão definidas todas as contas de utilizadores, privilégios de acesso ao *site*, empresas criadas e licenças geradas;

- Deve ser possível a configuração de todo o sistema em ambiente *intranet* para os clientes que pretendam ter todo o sistema em funcionamento dentro da sua empresa;
- Para restringir o acesso a determinadas funcionalidades e opções, devem ser criados dois tipos de licenças. A empresa que tenha uma licença normal não tem acesso à funcionalidade relatórios e à opção análise e tratamento de dados. A empresa que tenha a licença completa tem permissão de acesso a todas as funcionalidades e opções da aplicação *SCADA Web*.
- A aplicação *SCADA Web* deve permitir a integração do caso de estudo da *eneidaws* para monitorização e diagnóstico dos postos de transformação (caso de estudo explicado no capítulo 8). Para isso, a empresa pretende que o caso de estudo seja inserido na aplicação *SCADA Web* com funcionalidades semelhantes à aplicação *SCADA DSI*, com idioma em inglês;

Neste capítulo apresentou-se o funcionamento do sistema atual da *eneidaws*, e referiram-se as limitações que este sistema apresenta. Viu-se também os requisitos iniciais pretendidos pela empresa para o desenvolvimento deste novo projeto. Com base na informação deste capítulo, poder-se-á constatar posteriormente que as limitações existentes no sistema atual da *eneidaws* fazem sentido, tendo em conta o comparativo com alguns produtos *SCADA* existentes no mercado.

3. ESTADO DA ARTE

Neste capítulo apresenta-se uma perspetiva de quais os sistemas *SCADA* existentes atualmente, qual a sua origem e qual a importância destes sistemas hoje em dia para as empresas.

Procura-se ainda dar exemplos de alguns sistemas *SCADA* com o objetivo de conhecer quais as suas características principais e enumerar as suas especificidades e funcionalidades. A extração deste conjunto de funcionalidades primordiais permitiu evidenciar as deficiências existentes no sistema *SCADA* atual da *eneidaws* e comprovar que as funcionalidades pretendidas nos requisitos iniciais da empresa continuavam a ser insuficientes.

A informação reunida neste capítulo tornou-se fulcral para compreender os requisitos iniciais da empresa e para o refinamento dos requisitos propostos pelo aluno, para a elaboração da solução do sistema, como se poderá ver no capítulo 4.

3.1. O QUE SÃO SISTEMAS *SCADA*

Os sistemas *SCADA* (Webopedia, 2014) têm como objetivo supervisionar os processos produtivos nas empresas com o objetivo de melhorar a eficiência dos processos através da monitorização e controlo. Estes sistemas disponibilizam em tempo útil o estado atual do sistema, através de um conjunto de previsões, gráficos e relatórios, de modo a permitir a tomada de decisões operacionais apropriadas, quer automaticamente, quer por iniciativa do operador.

Inicialmente os sistemas *SCADA* permitiam informar periodicamente o estado do processo industrial, monitorizando sinais representativos de medidas e estados dos dispositivos, através de um painel de lâmpadas e indicadores, sem qualquer *interface* aplicacional com o utilizador. Com a evolução tecnológica, os computadores assumiram um papel importante na gestão, recolha e tratamento dos dados, tornando possível a visualização da informação no ecrã com recurso a *Human Machine Interface (HMI)* e a programação de regras para executar funções de controlo complexas.

Graças à evolução tecnológica dos computadores, os sistemas *SCADA* passaram a utilizar as tecnologias de computação e comunicação para automatizar a monitorização e controlo dos processos industriais. Com os computadores tornou-se possível efetuar a recolha de dados em ambientes complexos, eventualmente dispersos geograficamente, com apresentação gráfica de modo amigável para o utilizador, com recurso a *HMI*. Na figura 3 pode ver-se um exemplo de uma sala de controlo onde um ou mais operadores/utilizadores podem controlar todo o processo produtivo numa unidade industrial. Através do *interface* do sistema *SCADA*, o operador pode

supervisionar o estado de todos os equipamentos eletrônicos, como, por exemplo, máquinas industriais, unidades de controlo, sensores de vários tipos instalados no terreno em tempo real. O operador na sala de controlo pode saber se o processo produtivo se encontra em funcionamento correto ou se existe algum problema; desta forma, pode tomar decisões de supervisão para ajustar o sistema ou, sendo necessário, realizar intervenções técnicas no terreno, como, por exemplo, reparação de um ou mais equipamentos que se encontrem em funcionamento erróneo, afetando todo o processo industrial.



Figura 3 - Exemplo do HMI de um sistema SCADA numa sala de controlo (GECCAI,2014)

Na figura 4 apresenta-se um pequeno esquema, com o funcionamento de um sistema SCADA. Neste sistema simples, pretende-se controlar o fluxo de água no sistema através de uma bomba de água e como o nível de uma válvula. Para isso existem dois *Programmable Logic Controller (PLC)*: o *PLC1* é responsável por controlar a velocidade de funcionamento da bomba de água, e o *PLC2* controla o fluxo de água no sistema através do nível de uma válvula. Estes *PLC* estão programados para tomarem decisões de forma automática para manter o fluxo de água em valores nominais, garantindo a atividade do sistema de acordo com o funcionamento esperado; no entanto, o operador na sala de controlo pode supervisionar na aplicação SCADA todo o funcionamento do sistema em tempo real.

Possuindo o operador a capacidade de supervisionar todo o sistema, pode tomar várias decisões, como, por exemplo, alterar o valor do fluxo de água no circuito ajustando remotamente o valor referência nos *PLC*. Cada vez que é feito o ajuste nos valores de referência dos *PLC*, é alterada automaticamente a velocidade da bomba de água e o nível da válvula, sendo feito o ajuste ao fluxo de água no sistema para os valores pretendidos pelo operador. Para além de o operador poder realizar ajustes manuais no sistema, pode existir uma regra definida na aplicação SCADA que execute uma ação equivalente à do operador de forma automática, garantindo o funcionamento normal do sistema e libertando atenção do operador para controlar outros processos no sistema SCADA. Com este sistema SCADA, o operador

consegue ter a informação de todo o sistema de forma centralizada, verificando todos os equipamentos um a um, sem necessitar deslocar-se ao terreno.

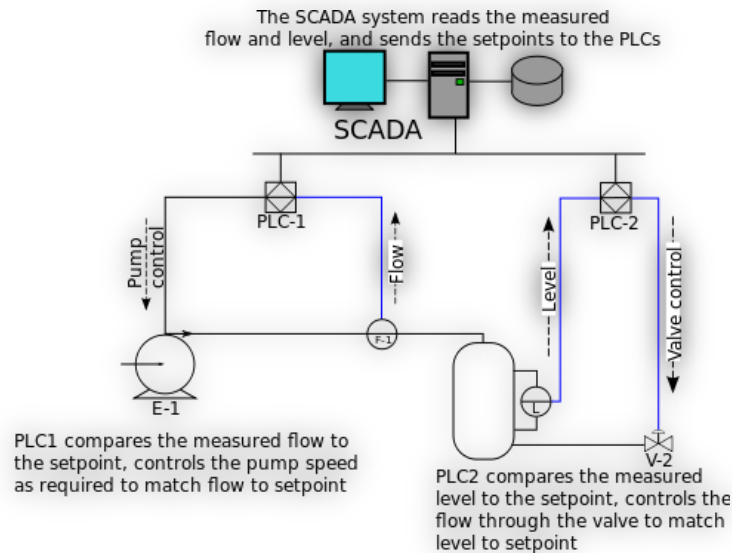


Figura 4 - Pequeno exemplo de um sistema SCADA (Wikipédia, 2012)

Hoje em dia, a aplicação destes sistemas *SCADA* é cada vez mais crucial na estrutura de gestão das empresas. Por este motivo, deixaram de ser vistos como meras ferramentas operacionais, ou de engenharia, e passaram a ser considerados uma importante fonte de informação. Atualmente estes sistemas estão presentes em áreas distintas, tais como a indústria de celulose, a indústria petrolífera, a indústria têxtil, a indústria metalúrgica, a indústria automóvel, a indústria eletrónica, a indústria química, entre outras.

Os elementos chave que as empresas podem esperar na utilização de sistemas *SCADA* são:

- **Qualidade:** A monitorização dos processos produtivos torna possível determinar níveis ótimos de trabalho. Caso estes níveis saiam dos valores esperados, o sistema *SCADA* pode gerar alarmes, alertando o operador para um eventual problema no processo produtivo. Podem ser feitas intervenções no processo de forma rápida, garantindo que os produtos finais terão sempre as características pretendidas no padrão de qualidade da empresa.
- **Redução dos custos operacionais:** No processo produtivo com inúmeros instrumentos de medição, seriam necessários muitos funcionários especializados para percorrer todo o processo produtivo a fim de realizar a leitura de todos os instrumentos, verificando-se que uma parte desta análise teria a probabilidade de erros humanos. Com os sistemas *SCADA*, é possível centralizar toda a informação de

todos os instrumentos em tempo real, sendo necessários poucos funcionários especializados, e com alguns cliques é possível realizar a leitura do processo industrial inteiro.

- **Maior desempenho da produção:** Com a disponibilidade de informação dos equipamentos no terreno em tempo real, podem ser feitas as intervenções necessárias de forma rápida. Com a rápida reparação dos equipamentos consegue-se ter uma maior disponibilidade do processo produtivo e, desta forma, um maior desempenho.
- **Base de informação para outros sistemas:** Os sistemas *SCADA* têm normalmente a capacidade de reunir os dados do processo produtivo e armazená-los em base de dados. Estes dados podem ser utilizados para gerar informações importantes que podem ser integradas com os sistemas de gestão da empresa em apoio às tomadas de decisão estratégicas da empresa.

Em ambientes industriais como estes, cada vez mais complexos e competitivos, os fatores relacionados com a disponibilidade e segurança da informação assumem elevada relevância, tornando-se necessário garantir que a informação está disponível de forma rápida e segura quando necessário, independentemente da localização geográfica. Torna-se, portanto, indispensável a implementação de mecanismos de acessibilidade, segurança e de tolerância a falhas, para que o ciclo de funcionamento destas empresas se mantenha constante.

3.2. EXEMPLOS DE APLICAÇÕES SCADA

Atualmente existem algumas aplicações *SCADA* que se caracterizam pelas componentes gráficas, pelo recurso a funcionalidades e pelas características modelares disponíveis. No entanto, a configuração destes sistemas é normalmente bastante complexa, sendo estas realizadas por especialistas. O sucesso do funcionamento destes sistemas depende integralmente destas configurações realizadas, caso contrário o sistema pode apresentar um funcionamento erróneo.

Grande parte destas aplicações contém uma série de protocolos de comunicação, como, por exemplo, RS-232¹³, MODBUS¹⁴, TCP/IP, CAN¹⁵, RS-485¹⁶, entre outros existentes. O uso de

¹³ **RS-232** (também conhecido por EIA RS-232C ou V.24) é um protocolo padrão para troca de dados binários na porta de série entre um terminal de dados, *Data Terminal Equipment (DTE)* e um comunicador de dados, *Data Communication Equipment (DCE)*.

¹⁴ **Modbus** é um protocolo de comunicação de dados utilizado nos sistemas de automação industrial, criado originalmente na década de 1970, pelo fabricante de equipamentos *Modicon*. É um dos mais antigos protocolos de comunicação usados em redes de controladores lógicos programáveis PLC, para aquisição de sinais.

¹⁵ **Controller Area Network (CAN)** é um protocolo de comunicação desenvolvido pela empresa alemã *Robert Bosch GmbH*, com base numa topologia de barramento para transmissão de mensagens em ambientes distribuídos.

diferentes protocolos de comunicação serve para que possa existir comunicação, controlo e supervisão dos mais variados autómatos existentes. Quanto maior o leque de protocolos numa aplicação *SCADA*, maior será o espectro de *hardware* (rede de sensores) com que esta aplicação pode comunicar, permitindo o controlo e a supervisão.

A existência destas bibliotecas com diferentes protocolos de comunicação facilita o trabalho das empresas que adquirirem estas aplicações, por serem genéricas, permitindo que estas funcionem com diferentes equipamentos. No entanto, os custos financeiros destes sistemas são muito elevados, tornando este género de produto inacessível às pequenas e médias empresas.

As aplicações *SCADA* podem ser caracterizadas em dois tipos: como propriedade das empresas que desenvolvem as suas soluções em *open source* disponível para todos os utilizadores, e empresas que pretendam usar estas aplicações com código aberto. A *eneidaws* tem como estratégia desenvolver o seu próprio *software* proprietário, tendo a sua aplicação servidor responsável por comunicar exclusivamente com *hardware* produzido por si e a aplicação *SCADA DSI*, responsável por supervisionar e monitorizar os dados adquiridos pelas redes de sensores por si implementadas.

Apresentam-se três exemplos de aplicações *SCADA*, onde são mostradas as características de cada uma das aplicações. Na secção 3.3 é feita uma análise comparativa das três aplicações, sendo retirado o leque de funcionalidades principais destes sistemas. Por fim é feita uma análise comparativa das funcionalidades principais destes sistemas com o sistema atual da *eneidaws*.

WinCC

Simatic WinCC (Siemens AG, 2014) é um sistema *SCADA*, com funções eficientes no controlo de processos automatizados. Com o *Simatic WinCC* são criados processos de visualização com funcionalidades completas para controlo e monitorização de todos os segmentos da indústria. Este *software* permite, ainda, a criação de interfaces *HMI* customizados de acordo com o sistema específico; pode ver-se o exemplo de um *interface* para monitorização dos sensores instalados em silos, *pipes*, entre outros, configurado neste *software*, e sua representação gráfica na figura 5.

¹⁶ **RS-485** é um protocolo padrão para comunicação serial, também denominado EIA-485 por ser desenvolvido pela *Electronics Industry Association (EIA)*. O padrão RS-485 é baseado na transmissão diferencial de dados, através de um par de fios, sendo ideal para transmissão em altas velocidades, longas distâncias e em ambientes propícios a interferência eletromagnética.

O *WinCC* é um sistema moderno com uma *interface user friendly* para uso no mundo industrial, oferecendo operações fiáveis e uma configuração eficiente. Em conjunto com o processo integrado de base de dados, o *WinCC* representa a troca de informação dentro da empresa e, devido à inteligência de visualização, traz consigo uma maior transparência à produção.

Assim, o *WinCC* é um *software SCADA* com informação disponível em tempo real, composto pelos seguintes módulos principais:

- *WinCC Explorer*: é o gestor dos módulos, utilizado sobretudo para aceder a outros módulos;
- *Data Manager (DM)*: através do módulo *DM* configuram-se as *tags* (etiquetas associadas a variáveis), que podem ser internas (sem ligação ao processo) e externas (com ligação ao processo). O processo de configuração das *tags* varia de *driver* para *driver*. O *DM* será o responsável por solicitar ao *driver* as *tags* necessárias para animar o *interface*, fazer gráficos, etc.;
- *Alarm Logging*: módulo para gestão de todo os alarmes existentes no sistema;
- *Tag Logging*: módulo para gerir e disparar as rotinas para armazenar as *tags* em períodos/situações previamente configuradas;
- *Graphic Designer*: módulo para configuração e edição das janelas do sistema;
- *Global Script*: módulo para edição de *Scripts*/Funções.

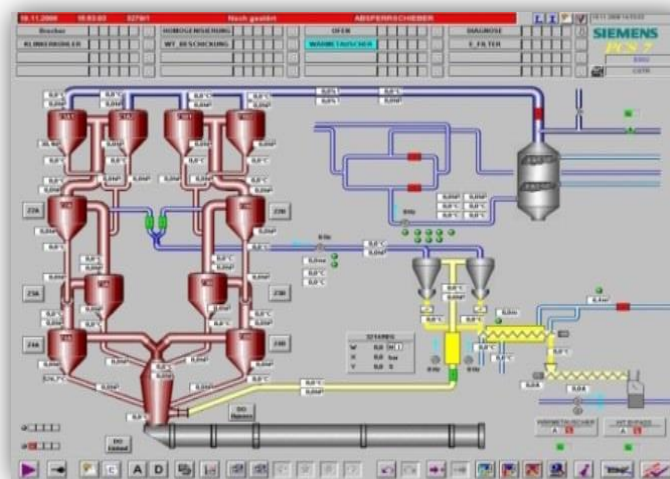


Figura 5 - Exemplo de monitorização de sensores no software WinCC (Siemens AG, 2014)

Movicon

O *Movicon* (*Monitoring, Vision, and Control*), é um *software SCADA* concebido para criar *interfaces* homem-máquina, baseado em *Personal Computer (PC)*. Na figura 6 apresenta-se um exemplo do editor de projetos desta aplicação.

A função do *Movicon* é supervisionar os processos de automação, usando *interfaces* animados, chamados *Synoptic Windows*. Esta aplicação permite controlar os processos produtivos através da utilização de *interfaces* denominados *dialog boxes*, bem como uma variada gama de outras funções que permitem o completo controlo do processo físico de um modo simples e seguro.

Esta aplicação permite uma fácil comunicação com o processo que é suposto interagir. Os componentes utilizados na gestão de processos, tais como *PLC*, controladores de temperatura, entre outros, constituem o sistema onde o *Movicon* está integrado, comunicando entre si através de portas de série, *modems*, redes *ethernet*, etc.

Este sistema pode ser usado por operadores em micro terminais, em *PC* móveis com *Windows CE*, como também pode ser usado em sistemas distribuídos cliente/servidor, interligado com *PLC*, redes industriais e redes de campo em grandes unidades industriais.

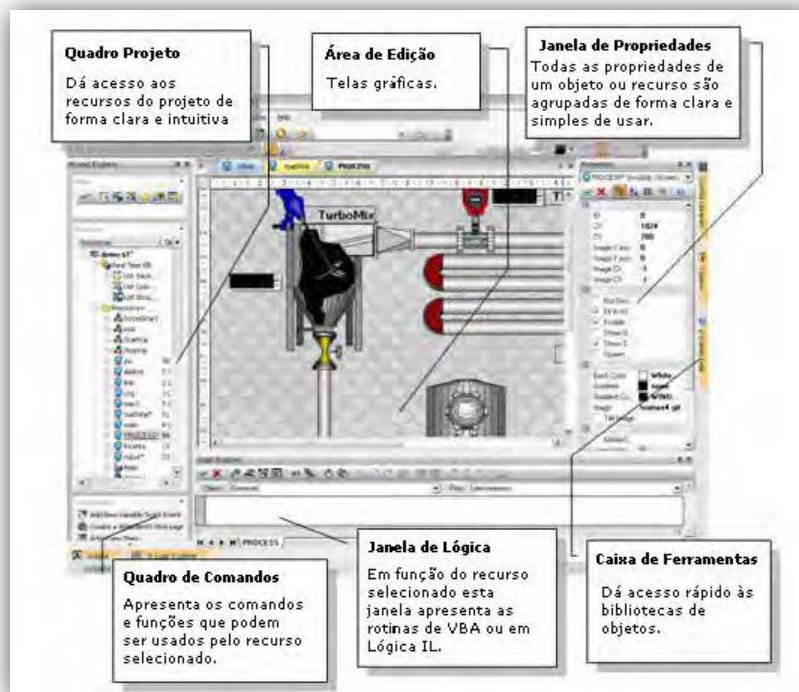


Figura 6 - Editor de projetos do Movicon X (Exata,2010).

Assim, o *Movicon X* é um *software SCADA* que apresenta as seguintes características:

- O *Movicon X* foi construído com tecnologias normalizadas para assegurar o seu investimento. Tecnologias como *eXtensible Markup Language (XML)*, *Open Database Connectivity (ODBC)*, *Object Linking and Embedding for Process Control (OPC)*¹⁷, *Visual Basic for Applications (VBA)*, *Simple Object Access Protocol (SOAP)*, *Web services*, *Transmission Control Protocol (TCP)*, *User Datagram Protocol (UDP)*, *HTTP* e *Structured Query Language (SQL)*, foram incorporadas no produto de forma a garantir o acesso fácil e transparente aos dados.
- O *software* apresenta ainda uma novidade interessante, a tecnologia *Web*, que no *Movicon X* surge renovada, devido à tecnologia *Java* que interage perfeitamente com *XML*, *SVG* e *Web Services*. A nova plataforma permite o acesso ao servidor a partir de *Browsers* de *Internet* em qualquer plataforma (*Windows*, *Linux*, *Palm* e alguns sistemas operativos de telemóveis) (Exata, 2010; Quintas, 2004).
- Permite criar e editar *interfaces*, para visualização do sistema configurado;
- Admite a visualização e gestão de alarmes e eventos, ocorridos em todo o sistema;
- Tem presente uma *Application Programming Interface (API)* baseada em *Web Service*, para acesso a informação e integração com outros sistemas.

ScadaBR

O *ScadaBR* é um sistema *SCADA open source*, para supervisão e controlo de processos produtivos, disponibilizado em licença gratuita. Completando 5 anos desde a primeira versão, o *ScadaBR* pode ser usado gratuitamente, estando à disposição toda a documentação e código fonte da aplicação, existindo ainda a possibilidade de alteração do código fonte e redistribuição do mesmo (ScadaBR, 2012). Este sistema conta com vários casos de uso nas áreas de automação de processos industriais, como redes de distribuição água e energia, automação predial e residencial, entre outros. Esta aplicação funciona com um espectro diversificado de sensores, por contar com suporte para mais de vinte protocolos de comunicação, o que permite a compatibilidade com uma variada gama de *hardwares* de centenas de fabricantes em todo o mundo. Por fim o *ScadaBR* conta ainda com uma camada de integração com *software* de

¹⁷ OPC é a sigla para “*OLE for Process Control*”, onde *OLE* significa “*Object Linking and Embedding*”. Este é o nome dado a uma interface padronizada de comunicação que foi criada na tentativa de minimizar os problemas relacionados com a inconsistência dos “*drivers*” de equipamentos industriais de diferentes fabricantes. A “*OPC Foundation*” é a organização responsável pelas normas que estabelecem as características disponíveis aos clientes dos equipamentos que possuem o padrão *OPC* (OPC Foundation, 2014).

terceiros, através de *web services*, possibilitando o acesso à informação e integração com outras soluções o que permite o uso de sistemas distribuídos. Na figura 7, apresenta-se um exemplo da *interface* da aplicação para monitorização de sensores, da aplicação *ScadaBR*, que monitoriza várias temperaturas, e indica o número de placas produzidas por dia no processo produtivo.

Assim, o *ScadaBR* é um *software SCADA* que contém as seguintes características:

- Vários protocolos de comunicação com *PLC*, sensores e dispositivos através de: *OPC*, *Modbus*, *RS232*, *TCP*, *American Standard Code for Information Interchange (ASCII)*, *Distributed Network Protocol, version 3 (DNP3)*, entre outros;
- Possibilidade de criar e editar janelas para visualização do sistema configurado;
- Visualização de gráficos e relatórios, relativamente aos dispositivos instalados e configurados;
- Controlo de acesso aos dados e à visualização de opções da aplicação, existindo diferentes níveis de privilégios para as contas de utilizadores criadas para esta aplicação;
- Possibilidade de programação de regras, para dispositivos instalados através de *scripting*;
- Visualização e gestão de alarmes e eventos, existentes em todo o sistema;
- *API Web Service* para acesso a informação e integração com outros sistemas.



Figura 7 - Exemplo de monitorização de sensores no software *ScadaBR* (*ScadaBR*, 2012)

3.3. ANÁLISE COMPARATIVA DOS SISTEMAS SCADA

Da análise realizada às aplicações *SCADA*, considerando as características analisadas de cada uma das aplicações e com opiniões semelhantes (Costa, 1995; Costa, 2009), constata-se que os sistemas *SCADA* podem definir-se pelos seguintes conjuntos principais de funcionalidades:

- *Interface Web* simples com utilizador, capaz de representar o sistema a monitorizar;
- Controlo de acessos, limitando determinadas opções e informações ao utilizador, com base nos diferentes níveis de privilégios;
- Possibilidade de monitorizar toda a informação do sistema;
- Visualização da informação do sistema através de animações, sons, padrões e cores (por exemplo, a existência de um alarme deve ser visível de forma clara ao utilizador, por variação de cores ou sons);
- Possibilidade de configurações gráficas de sinópticos ou animações;
- Gestão e configuração de regras para alarmes e eventos;
- Armazenamento de dados adquiridos em base de dados;
- Possibilidade de analisar histórico de dados recolhidos, através de gráficos, tabelas e pesquisa de informação customizada pelo utilizador;
- *Interfaces* para configuração de redes, equipamentos, unidades de controlo, sensores, entre outros;
- Tratamento de dados com possibilidade de exportar para Excel, *Portable Document Format (PDF)*, entre outros formatos;
- Disponibilização de informação sobre o sistema via *Web Service*.
- Utilização de diferentes protocolos de comunicação, para monitorização de diferentes equipamentos;
- As aplicações apresentam-se como sendo escaláveis;
- Possibilidade de integração em rede;
- Processos cliente/servidor distribuídos.

3.3.1. Análise Comparativa do Sistema SCADA atual da *eneidaws*

Com base na análise comparativa entre as aplicações SCADA apresentadas anteriormente, pode extrair-se um conjunto de características principais de funcionalidades existentes nas aplicações deste género. Esta análise é útil para compreender os requisitos pretendidos pela empresa e para identificar novas funcionalidades e opções que o novo sistema SCADA da *eneidaws* deve apresentar.

As características apresentadas e a análise efetuada à atual aplicação SCADA da *eneidaws*, onde foram expostas as limitações desta aplicação, vêm a comprovar as necessidades evidenciadas pela *eneidaws* e a razão da existência deste projeto. Em concreto, a aplicação SCADA DSI não apresenta determinadas funcionalidades que são importantes para um sistema deste género, tais como: controlo de acessos; configuração de regras para alarmes; pesquisa de informação customizada e tratamento de dados; exportação de dados para *PDF* e *Excel*; visualização de alarmes na aplicação com recurso a cores e a animações, entre outras. Todas estas funcionalidades e elementos, referidos neste capítulo, têm uma grande importância para o refinamento da análise de requisitos.

Neste capítulo explicou-se o que são os sistemas SCADA e a sua importância para as empresas atualmente. Analisou-se um conjunto de aplicações SCADA existentes no mercado, tornando-se possível identificar um conjunto funcionalidades e opções comuns entre estas. Com base nestas funcionalidades, em grande parte inexistentes no sistema SCADA da *eneidaws*, confirmou-se a importância na existência deste projeto. As funcionalidades basilares normalmente presentes nas aplicações SCADA foram tidas em conta no *brainstorming*, como um refinamento aos requisitos iniciais da empresa, para a elaboração final dos requisitos propostos e para proposta da solução do sistema, como se poderá comprovar no capítulo 4.

4. ANÁLISE DE REQUISITOS E SOLUÇÃO PROPOSTA

Neste capítulo, apresenta-se a solução proposta pelo aluno para o desenvolvimento deste projeto, tendo em conta os requisitos iniciais da empresa e as funcionalidades principais geralmente existentes nos sistemas *SCADA* presentes no mercado. Com estes dados tornou-se possível elaborar uma solução para o novo sistema e os respetivos requisitos necessários validados com a *eneidaws*, tendo em conta as vantagens previstas da solução proposta para implementação do novo sistema.

4.1. DESCRIÇÃO DA SOLUÇÃO PROPOSTA

A solução proposta contempla um novo sistema *SCADA* para a *eneidaws* poder supervisionar e controlar as diferentes redes de sensores instaladas. Como base da elaboração desta proposta e dos requisitos definidos para este projeto, teve-se em conta os requisitos iniciais pretendidos pela empresa, vistos no capítulo 2, e as funcionalidades essenciais existentes nas aplicações *SCADA* presentes no mercado, vistas no capítulo 3. Com esta informação compilada, considerou-se que o desenvolvimento deste projeto deveria contemplar quatro novas aplicações. Estas aplicações têm de apresentar mais funcionalidades e opções, assentes em tecnologias *Windows* com arquitetura de desenvolvimento modular e a elaboração de um novo modelo para gestão de base de dados, capaz de armazenar toda a informação do novo sistema *SCADA*. Contemplou-se a existência destas quatro aplicações para existir separação de responsabilidades de cada uma delas no novo sistema *SCADA*: por um lado tem-se aplicações que são serviços e que têm a responsabilidade de receber informações das redes de sensores, criando-se abstração entre a camada de *hardware* e do *software*, no caso através da aplicação servidor que recebe informações das redes de sensores e da aplicação serviço de alarmes que irá estar continuamente a verificar a existência de alarmes; por outro lado tem-se funcionamento passivo da aplicação *Web SCADA*, que servirá para consulta, controlo e tratamento de informações relativamente às redes de sensores configuradas no sistema, e, por fim, apresenta-se a aplicação *Web Service* que disponibilizará informações sobre as redes de sensores via *HTTP*.

As quatro aplicações a serem desenvolvidas para o novo sistema *SCADA* serão as seguintes:

- Uma aplicação servidor responsável por receber todos os dados adquiridos pelas redes de sensores instaladas no terreno e guardar todos os dados em base de dados;
- Uma aplicação serviço de alarmes responsável exclusivamente por notificar os utilizadores da ocorrência de alarmes nas redes de sensores por *e-mail* e *SMS*;

- Uma aplicação baseada em *Web Service* que permita a disponibilização de informação das redes de sensores através de pedidos *HTTP* e permita a integração com outros sistemas;
- uma nova aplicação *SCADA Web*, que permita configurar todas as redes de sensores e supervisionar toda a informação relativa a histórico de dados adquiridos pelas redes de sensores.

O novo sistema, ao contemplar aplicações com mais funcionalidades e opções, permite a versatilidade para integrar vários projetos que a *eneidaws* possa vir a ter, e em específico a integração do caso de estudo da empresa (capítulo 8), que tem como finalidade a monitorização dos postos de transformação energéticos, a nível ambiental, energético, de segurança, de qualidade do serviço, entre outros.

4.2. REQUISITOS FINAIS PARA NOVO SISTEMA

O levantamento de todos os requisitos para o novo sistema a desenvolver foi realizado através de entrevistas na empresa, observação direta e pela análise de funcionalidades e opções que existem nos sistemas *SCADA*. Para a representação dos requisitos definidos para o novo sistema recorreu-se aos casos de uso, que constituem a técnica em *Unified Modeling Language (UML)*, para representação dos requisitos (Nunes e O'Neill, 2004).

4.3. ATORES

Apresenta-se de uma forma geral os vários atores envolvidos no sistema, e explica-se o que cada um representa:

- **Administrador Geral** – Este ator representa o administrador geral da *eneidaws*; será este o utilizador que terá como responsabilidade a configuração, o suporte e a manutenção de todo o sistema, tendo neste caso o acesso total a todas as aplicações;
- **Cliente Administrador** – Este ator representa o cliente da *eneidaws* que corresponde ao administrador da sua própria empresa, o qual tem acesso à aplicação *SCADA web*, com privilégios de acesso elevados para configurações, mas nunca superiores aos do Administrador Geral;
- **Cliente Utilizador** – Este ator terá acesso à aplicação *Web SCADA*, e representa o utilizador funcionário da empresa do Cliente Administrador. Os privilégios de acesso são baixos, possibilitando apenas a visualização e consulta de informação;

- **Anónimo** – Este ator representa o utilizador anónimo que pode visualizar a parte pública da aplicação *SCADA Web*;
- **Rede de Sensores** – Este ator representa as redes de sensores que adquirem os dados das unidades industriais e submetem esta informação para o servidor onde posteriormente essa informação é guardada em base de dados;
- **Aplicação *SCADA Web*** – Este ator representa a aplicação *SCADA Web*, que serve para consulta dos dados adquiridos das redes de sensores. Pontualmente, este ator submete mensagens ao servidor para este encaminhar para as redes de sensores.

4.4. REQUISITOS DA APLICAÇÃO SERVIDOR

Esta aplicação servidor tem a função de comunicar com as redes de sensores instaladas nas unidades industriais. De acordo com as necessidades evidenciadas pela empresa, considerou-se que os requisitos (tabela 4) necessários para desenvolvimento desta aplicação deveriam assentar nos seguintes pontos:

- Deve manter-se a mesma estrutura de mensagens de dados usada atualmente pela *eneidaws*, de forma a garantir que esta aplicação continua a funcionar com os atuais equipamentos produzidos, instalados em diferentes redes sensores;
- A aplicação deverá ainda ter capacidade de analisar todas as mensagens recebidas e notificar a ocorrência de alarmes, sempre que se verifiquem, por *SMS* e *e-mail*;
- A funcionalidade para validação de alarmes deverá ser opcional, permitindo ao ator a escolha de ativar ou desativar esta funcionalidade.

4.4.1. Requisitos Funcionais

Tabela 4 - Requisitos aplicação servidor

Código ¹⁸	Requisito	Descrição
RF1AS	Ligar Servidor	Deve obrigar o ator administrador geral a configurar o <i>IP</i> e porto do servidor, antes que as comunicações fiquem ativas. Opcionalmente o ator deve ativar ou desativar a validação de alarmes nas configurações do servidor.

¹⁸ RF1AS – Requisito Funcional número 1, Aplicação Servidor

RF2AS	Submete Mensagem	A rede de sensores submete os dados adquiridos para o servidor. O servidor deve ser capaz de tratar todas as mensagens recebidas pela rede de sensores.
RF3AS	Submete Ação Mensagem	Sempre que a aplicação <i>web</i> pretende executar uma ação numa rede de sensores submete uma mensagem para o servidor. O servidor deve ser capaz de tratar todas as mensagens recebidas pela aplicação <i>web</i> , encaminhando esta mensagem/ação para a rede de sensores.
RF4AS	Tratar Mensagens	O servidor que deve ser capaz de receber e enviar todas as mensagens.
RF5AS	Guardar Mensagem Base de Dados	Todas as mensagens recebidas e enviadas pelo servidor devem ser guardadas nas bases de dados correspondentes a cada cliente da <i>eneidaws</i> .
RF6AS	Verificar Alarmes	O ator deve opcionalmente ter possibilidade de ativar ou desativar a verificação de alarmes. Se a opção estiver ativa, todas as mensagens recebidas pela rede de sensores devem ser validadas quanto à existência de alarmes. Se a opção estiver desativada não devem ser verificados alarmes.
RF7AS	Enviar <i>e-mail</i> ou <i>SMS</i>	Devem ser enviados alarmes por <i>SMS</i> e/ou <i>e-mail</i> , se a opção verificar alarmes estiver ativa, cada vez que existir um alarme. Se a opção verificar alarmes estiver desativada, não devem ser enviados alarmes por <i>e-mail</i> nem <i>SMS</i> .
RF8AS	Desligar Servidor	Devem ser terminados todos os processos de comunicação que estejam a decorrer, se o servidor for desligado.
RF9AS	Escolher Opção	Deve listar todas as opções entre as quais o ator possa escolher.
RF10AS	Enviar Alive ID	Deve permitir ao ator enviar mensagens às redes conectadas ao servidor, “perguntando” quais são os equipamentos que se encontram ativos nessas redes.
RF11AS	Enviar Mensagem Teste	Deve servir para o ator enviar mensagens de teste para um equipamento numa rede.
RF12AS	Listar Clientes Conectados	Deve servir para o ator listar todas as redes ligadas ao servidor.
RF13AS	Ligar Log	Deve servir para ligar o <i>Log</i> que permite visualizar as mensagens que estão a ser trocadas com o servidor.
RF14AS	Desligar Log	Deve servir para desligar o <i>Log</i> que permite visualizar as mensagens que estão a ser trocadas com o servidor.
RF15AS	Limpar Log	Deve servir para limpar o <i>Log</i> de mensagens que estão a ser trocadas com o servidor.
RF16AS	Consultar Ajuda	Deve servir para o ator administrador geral consultar o manual de ajuda na aplicação.

4.4.2. Diagrama de Casos de Uso

No diagrama de casos de uso visto na figura 8, demonstra-se de forma esquemática os requisitos da aplicação servidor (tabela 4) e os atores que interagem com esta aplicação. Os atores Rede de Sensores e Aplicação SCADA Web têm a possibilidade de fazer submissão de mensagens de dados ao servidor; por sua vez, este responsabiliza-se por verificar o conteúdo das mensagens e guarda as mesmas em base de dados. O ator Administrador Geral tem possibilidade de ligar o servidor, mas para isso tem de definir obrigatoriamente um IP, porto, e, se o pretender, ativar ou desativar a verificação de alarmes, para que a aplicação entre em funcionamento. Este ator tem ainda a possibilidade de escolha de várias opções, como: desligar a aplicação, consultar o manual de ajuda, enviar mensagens de testes, entre outros.

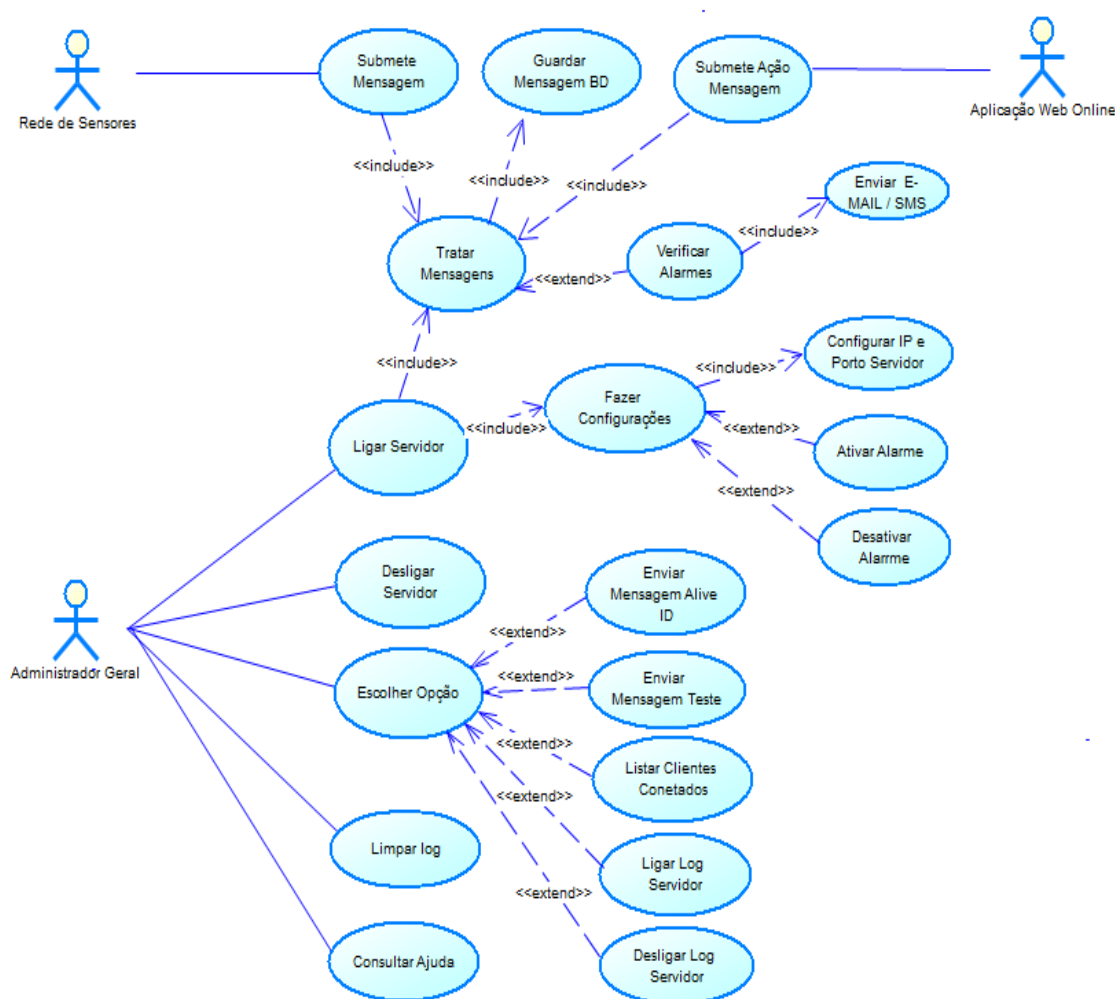


Figura 8 - Diagrama de Casos de Uso Aplicação Servidor

4.4.3. Requisitos Não Funcionais

Os requisitos não funcionais, considerados para a aplicação servidor, são os seguintes, como se pode ver na tabela 5.

Tabela 5 - Requisitos não funcionais aplicação servidor

Código ¹⁹	Requisito Não Funcional
RNF1AS	Usabilidade - A aplicação deve ter um <i>interface</i> simples e amigável.
RNF2AS	Performance - O processamento de cada mensagem recebida no servidor não deve ser superior a 60 segundos.
RNF3AS	Escalabilidade - O servidor deve ser capaz de suportar conexão de várias redes de sensores e tratar mensagens recebidas.
RNF4AS	Disponibilidade – O servidor deve ter capacidade de adaptação quando um requisito muda.
RNF5AS	O idioma da aplicação deve ser o português.
RNF6AS	A aplicação deve ter uma arquitetura de desenvolvimento em várias camadas.
RNF7AS	Deve ser possível configurar em <i>eXtensible Markup Language (XML)</i> a <i>string</i> de conexão à base de dados <i>SQL Server</i> .
RNF8AS	A aplicação deve ser desenvolvida com tecnologia <i>Windows Forms</i> .

4.5. REQUISITOS DA APLICAÇÃO SERVIÇO DE ALARMES

Esta aplicação Serviço de Alarmes tem a função de executar exclusivamente a verificação da existência de alarmes através da consulta da base de dados e notificar os utilizadores através de *SMS* e *e-mail*.

De acordo com as necessidades evidenciadas pela empresa, considerou-se que os requisitos (tabela 6) para desenvolvimento da aplicação deveriam assentar nas seguintes funcionalidades:

- A possibilidade de esta aplicação ter um relógio que permitisse definir a periodicidade da verificação da existência de alarmes;
- A hipótese de notificar o utilizador por *e-mail* e *SMS*.

¹⁹ **RNF1AS** – Requisito Não Funcional número 1, Aplicação Servidor

4.5.1. Requisitos Funcionais

Tabela 6 - Requisitos Aplicação Serviço de Alarmes

Código ²⁰	Requisito	Descrição
RF1ASA	Ligar Serviço Alarmes	Deve obrigar o ator administrador geral a configurar o relógio para verificação de alarmes em segundos, minutos ou dias, i.e., a consulta da tabela de alarmes é feita periodicamente pelo servidor de alarmes; por exemplo, de 60 em 60 segundos.
RF2ASA	Verificar Alarmes	Deve permitir que seja feita a verificação da existência de alarmes periodicamente com base no relógio configurado, e caso seja encontrado alarme é notificado o utilizador por <i>e-mail</i> e/ou <i>SMS</i> .
RF3ASA	Enviar <i>e-mail</i> e/ou <i>SMS</i>	Cada vez que existirem alarmes, estes devem ser enviados por <i>SMS</i> e/ou <i>e-mail</i> .
RF4ASA	Desligar Serviço Alarmes	Sendo desligado o serviço de alarmes devem ser terminados todos os processos para verificação de alarmes.
RF5ASA	Consultar Ajuda	Deve servir para o ator administrador geral consultar o manual de ajuda na aplicação.

4.5.2. Diagrama de Casos de Uso

No diagrama de casos de uso visto na figura 9, demonstra-se de forma esquemática os requisitos da aplicação serviço de alarmes (tabela 6) e o ator com acesso a esta aplicação. O ator, para colocar a aplicação em funcionamento, precisa configurar o relógio para verificação e notificação de alarmes. Este ator tem ainda a possibilidade de desligar a aplicação e consultar o manual de ajuda.

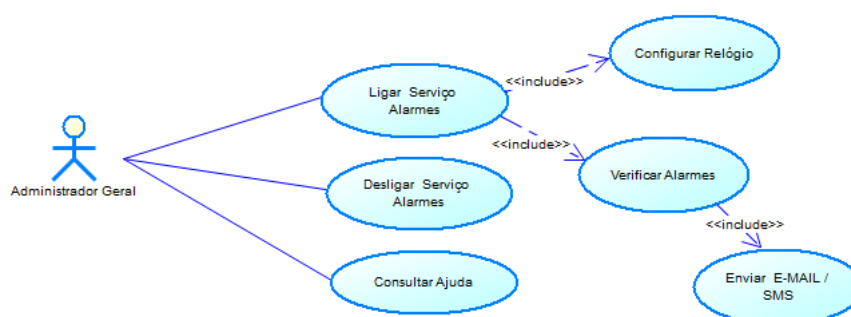


Figura 9 - Diagrama de Casos de Uso Aplicação Serviço de Alarmes

²⁰ RF1ASA – Requisito Funcional número 1, Aplicação Serviço de Alarmes

4.5.3. Requisitos Não Funcionais

Os requisitos não funcionais, considerados para a aplicação serviço de alarmes, são os seguintes, como se pode ver na tabela 7.

Tabela 7 - Requisitos não funcionais aplicação serviço de alarmes

Código ²¹	Requisito Não Funcional
RNF1ASA	Usabilidade – A aplicação deve ter um <i>interface</i> simples e amigável.
RNF2ASA	Performance – O processamento de cada alarme por <i>e-mail</i> ou <i>SMS</i> não deve ser superior a 60 segundos após verificado o alarme.
RNF3ASA	Escalabilidade – A aplicação serviço de alarmes deve ser capaz de processar vários alarmes em simultâneo.
RNF4ASA	Disponibilidade – A aplicação deve ter capacidade de adaptação quando um requisito muda.
RNF5ASA	A aplicação deve ter uma arquitetura de desenvolvimento em várias camadas.
RNF6ASA	O idioma da aplicação deve ser o português.
RNF7ASA	Deve ser possível configurar em <i>eXtensible Markup Language (XML)</i> a <i>string</i> de conexão à base de dados <i>SQL Server</i> .
RNF8ASA	A aplicação deve ser desenvolvida com tecnologia <i>Windows Forms</i> .

4.6. REQUISITOS DA APLICAÇÃO WEB SERVICE

Esta aplicação baseada em *Web Service* serve para disponibilizar informação através da internet em qualquer parte do mundo por pedidos *HTTP*, *GET* e *POST* relativamente às redes de sensores, com o objetivo da partilha de informação sobre as mesmas e facilitando a integração com outros sistemas.

De acordo com as necessidades evidenciadas pela empresa, considerou-se que os requisitos (tabela 8) para desenvolvimento desta aplicação deveriam assentar nas seguintes funcionalidades:

²¹ RNF1ASA – Requisito Não Funcional número 1, Aplicação Serviço de Alarmes

- Vários pedidos *GET*, para obtenção de diversas informações sobre as redes de sensores;
- Um pedido *POST*, para submissão de dados pela rede de sensores para a aplicação *Web Service*.

4.6.1. Requisitos Funcionais

Tabela 8 - Requisitos aplicação *Web Service*

Código ²²	Requisito	Descrição
RF1AWS	Valida Chave <i>Web Service</i>	Deve ser validada a chave única do ator que usa o <i>Web Service</i> .
RF2AWS	Obter todas as localizações	Deve ser disponibilizada toda a informação em <i>JavaScript Object Notation (JSON)</i> , relativamente a todas as localizações.
RF3AWS	Obter todas as Redes	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todas as redes.
RF4AWS	Obter todos os Sensores	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todos os sensores.
RF5AWS	Obter todas Redes de uma localização	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todas as redes de uma localização.
RF6AWS	Obter todos os Grupos	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todos os grupos.
RF7AWS	Obter todos os Sensores de um Grupo	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todos os sensores de um grupo.
RF8AWS	Obter todos os Modelos	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todos os modelos.
RF9AWS	Obter todos os Sensores de um Modelo	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todos os sensores de um modelo.
RF10AWS	Obter todas as Entradas de um Sensor	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todas as entradas de um sensor.

²² RF1AWS – Requisito Funcional número 1, Aplicação *Web Service*

Código	Requisito	Descrição
RF11AWS	Obter todas as Mensagens de uma Entrada	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todas as mensagens de dados de uma entrada.
RF12AWS	Obter todas as Mensagens	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todas as mensagens de dados.
RF13AWS	Obter intervalo de Mensagens	Deve ser disponibilizada toda a informação em <i>JSON</i> , relativamente a todas as mensagens de um intervalo (por exemplo, as 15 últimas mensagens recebidas).
RF14AWS	Submete Mensagem	Deve ser recebida toda a informação em <i>JSON</i> , de uma determinada mensagem. Esta deve ser validada se a mensagem referente à entrada de um sensor está em alarme. Caso esteja, deve ser enviado um alarme por <i>e-mail</i> e <i>SMS</i> . Esta mensagem é enviada pela rede de sensores.
RF15AWS	Tratar Mensagens	O <i>Web Service</i> deve ser capaz de receber mensagens da rede de sensores.
RF16AWS	Guardar Mensagem Base de Dados	O <i>Web Service</i> deve ser capaz de guardar todas as mensagens recebidas em base de dados.
RF17AWS	Verificar Alarmes	O ator deve ter a possibilidade de ativar ou desativar verificação de alarmes. Se a opção estiver ativa, todas as mensagens recebidas pela rede de sensores devem ser validadas da existência de alarme. Se a opção estiver desativada, não devem ser verificados os alarmes.
RF18AWS	Enviar <i>e-mail</i> e/ou <i>SMS</i>	Se a opção verificar alarmes estiver ativa, cada vez que existir um alarme devem ser enviados alarmes por <i>SMS</i> e/ou <i>e-mail</i> .

4.6.2. Diagrama de Casos de Uso

No diagrama de casos de uso da figura 10, demonstra-se de forma esquemática os requisitos da aplicação *Web Service* (tabela 8) e os atores que têm acesso a esta aplicação. Com este diagrama pode ver-se os pedidos que os atores Cliente Administrador e Administrador Geral podem fazer ao *Web Service* relativamente a dados e informações das redes de sensores. Por fim, é visível o ator Rede de Sensores, submetendo dados das redes de sensores instaladas no terreno. Posteriormente os dados submetidos por *Web Service* são analisados quando à ocorrência de alarmes e guardados em base de dados.

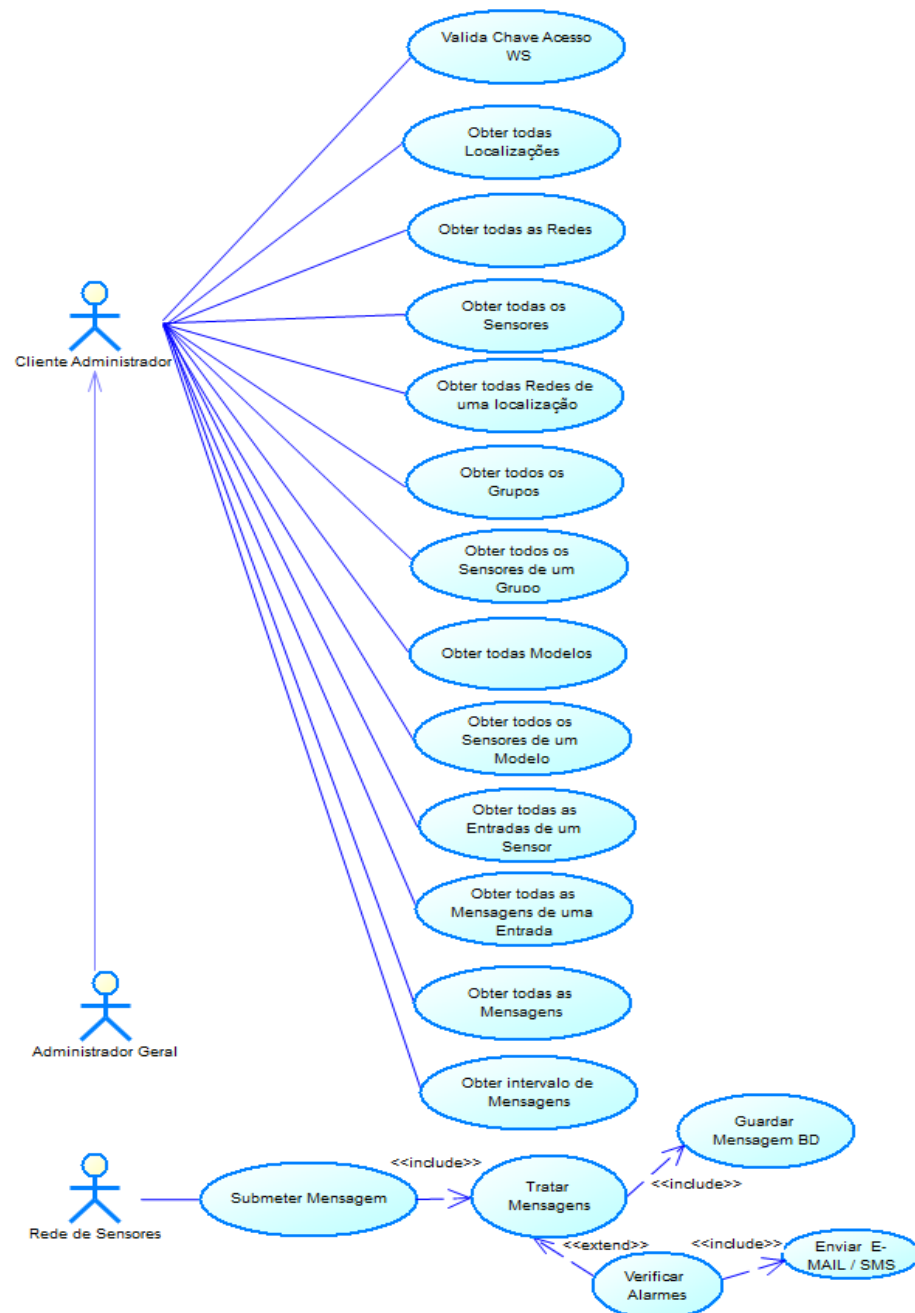


Figura 10 - Diagrama de Casos de Uso Web Service

4.6.3. Requisitos Não Funcionais

Os requisitos não funcionais, considerados para a aplicação *Web Service*, são os seguintes, como se pode ver na tabela 9.

Tabela 9 - Requisitos não funcionais aplicação Web Service

Código ²³	Requisito Não Funcional
RNF1AWS	Performance – Nas mensagens recebidas, quando existe alarme, o processamento dos alarmes por <i>e-mail</i> ou <i>SMS</i> não deve ser superior a 60 segundos após verificado o alarme.
RNF2AWS	Escalabilidade – A aplicação <i>Web Service</i> deve ser capaz de processar vários pedidos de informação em simultâneo.
RNF3AWS	Disponibilidade – A aplicação <i>Web Service</i> deve ter capacidade de adaptação quando um requisito muda; a aplicação <i>Web Service</i> deve estar disponível para o utilizador de forma permanente.
RNF4AWS	Segurança – Só utilizadores com chave válida podem ter acesso ao <i>Web Service</i> .
RNF5AWS	A aplicação <i>Web Service</i> deve ser alojada em sistemas operativos <i>Windows</i> .

4.7. REQUISITOS DA APLICAÇÃO SCADA WEB

Esta aplicação *Web* representa a aplicação *SCADA* para supervisão e controlo de todas as redes de sensores instaladas nas diferentes unidades industriais. De acordo com as necessidades evidenciadas pela empresa, considerou-se que os requisitos (tabela 10) para desenvolvimento desta aplicação deveriam assentar nas seguintes funcionalidades:

- A aplicação deve ser desenvolvida em ambiente *Web*, permitindo que este sistema possa funcionar em diferentes plataformas, bastando para isso o uso de um *browser* para acesso à aplicação;
- O ambiente gráfico da aplicação deve ajustar-se à área do dispositivo (*desktop*, *laptop*, *tablet*, *smartphone*) que pretenda visualizar a aplicação;
- Deve existir controlo de acessos para garantir a segurança da aplicação. Cada ator terá uma conta de acesso para fazer *login* na aplicação. Esta conta terá três níveis de acesso, representados pelos atores Administrador Geral, Cliente Administrador e Cliente Utilizador;
- Deve ser possível saber todas as ações que foram executadas por determinado ator através da sua conta de *login*;
- Esta aplicação deve permitir criar e gerir empresas, criar e gerir licenças de acesso à aplicação *Web* associada a uma empresa, e criar e gerir utilizadores associados a uma empresa;

²³ RNF1AWS – Requisito Não Funcional número 1, Aplicação *Web Service*

- A visualização das redes de sensores deve ser feita através de coordenadas *Global Positioning System (GPS)*, correspondentes a um ponto no mapa. Neste mapa deve ser possível saber o estado da rede (por exemplo: se existe alguma ocorrência de alarme dentro de uma rede, o ponto no mapa será vermelho ou amarelo; se não existir alarme, o ponto será verde). Todos os pontos que representam uma rede no mapa devem ter um *link* para visualização dos últimos dados adquiridos por uma rede de sensores;
- Esta aplicação deve permitir criar e gerir redes de sensores e importar ou exportar as redes de sensores automaticamente;
- A visualização do histórico de dados das redes de sensores e histórico de alarmes deve ser feito através de gráficos e tabelas;
- Deve ser possível a pesquisa de informação específica relativa às redes de sensores e exportar os dados para formato *Excel*, *PDF*, entre outros;
- Deve ser possível criar e gerir regras para alarmes, e associar utilizadores responsáveis por receber esses alarmes, para além de visualizar alarmes ocorridos por *e-mail*, *SMS* e respetiva descrição na aplicação *Web*;
- A aplicação deve ter uma área de suporte que pode ser usada pelos utilizadores para esclarecer dúvidas com o administrador da empresa e este esclarecer dúvidas com o administrador geral da aplicação;
- A aplicação deve ser desenvolvida com duas bases de dados: da base de dados geral constam as empresas, licenças e utilizadores associados a essas empresas; na segunda base de dados são guardados os dados adquiridos por todas as redes referentes a um cliente, i.e., cada cliente deve ter uma base de dados para guardar os dados das suas redes.

4.7.1. Requisitos Funcionais

Tabela 10 - Requisitos Gerais Aplicação SCADA Web

Código ²⁴	Requisito	Descrição
RF1ASW	Fazer Login	Deve obrigar o ator anónimo a inserir o seu <i>login</i> e <i>password</i> , para poder entrar na aplicação.
RF2ASW	Recuperar <i>Password</i>	Deve permitir ao ator anónimo fazer recuperação da <i>password</i> , sempre que o mesmo tenha uma conta de acesso válida por <i>e-mail</i> .

²⁴ RF1ASW – Requisito Funcional número 1, Aplicação SCADA Web

Código	Requisito	Descrição
RF3ASW	Ver Contactos	Deve permitir o ator anónimo ver os contactos da empresa <i>eneidaws</i> .
RF4ASW	<i>Dashboard</i>	Deve conter notificações dos alarmes críticos, alertas e os <i>links</i> para as outras opções da aplicação. Os <i>links</i> das opções ficam visíveis de acordo com as permissões de cada ator, Cliente Utilizador, Cliente Administrador e Administrador Geral.
RF5ASW	Empresa	Deve conter todo os dados da empresa visíveis para o ator Cliente Administrador.
RF6ASW	Informação Pessoal	Deve conter todo os dados referentes ao ator logado no sistema.
RF7ASW	Rede Sensores	Deve conter toda a informação para visualização das redes de sensores num mapa por georreferenciação e numa tabela. Aqui poderá ser escolhida a visualização dos últimos dados adquiridos numa determinada rede, sabendo o estado de cada sensor. Esta opção está disponível para os atores Cliente Utilizador, Cliente Administrador e Administrador Geral.
RF8ASW	Alarmes	Deve permitir consultar a informação dos alarmes críticos e alertas adquiridos e as informações associadas aos mesmos. Este requisito está disponível para os atores Cliente Utilizador, Cliente Administrador e Administrador Geral. O ator Cliente Administrador pode validar alarme como tratado e colocar uma descrição, após verificar a sua ocorrência.
RF9ASW	Reporte	Deve permitir consultar o histórico dos dados adquiridos em gráficos, tabelas e a possibilidade de exportar estas informações para formato Excel, <i>PDF</i> , entre outros. Deve ainda ser possível fazer a pesquisa de informação específica de sensores numa rede.
RF10ASW	Suporte	Deve permitir ao ator Cliente Utilizador tirar dúvidas relativas à aplicação <i>web</i> com o ator Cliente Administrador. O ator Cliente Administrador deve ter a possibilidade de tirar dúvidas com o ator Administrador Geral.
RF11ASW	Configurações	Deve permitir ao ator Cliente Administrador: criar e gerir redes de sensores; criar e gerir regras para alarmes e associar utilizadores que recebem alarmes; criar e gerir os seus utilizadores e atribuir níveis de privilégios; importar e exportar redes de sensores automaticamente; visualizar erros ocorridos na configuração da rede de sensores.
RF12ASW	Administração	Deve permitir ao ator Administrador Geral: criar e gerir empresas; criar e gerir licenças atribuídas a uma empresa; criar e gerir utilizadores associados a uma empresa.
RF13ASW	Consultar Ajuda	Deve permitir aos atores consultar o manual de ajuda da aplicação.
RF14ASW	Terminar Sessão	Deve permitir ao ator logado fazer <i>logout</i> , terminando a sessão e saindo da aplicação.

4.7.2. Diagramas de Casos de Uso

No diagrama de casos de uso, visto na figura 11, demonstra-se de forma esquemática os requisitos (tabela 10) do ator anónimo. Este ator tem a possibilidade de fazer *login* na aplicação *SCADA Web*; se perder a *password*, pode recuperar a mesma por *e-mail*; e, por fim, visualizar os contactos da empresa *eneidaws*.

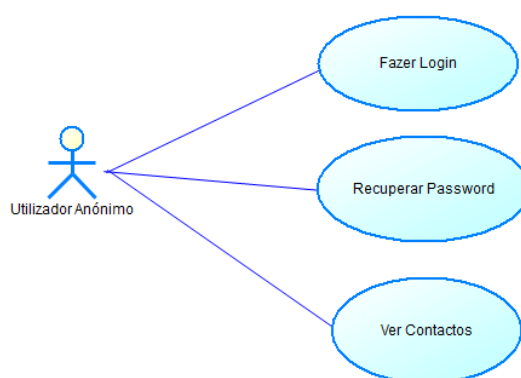


Figura 11 - Diagrama de Casos de uso Ator Anónimo Aplicação SCADA Web

O diagrama de pacotes e casos de uso da figura 12 representa de forma esquemática os requisitos gerais das respetivas funcionalidades da aplicação *SCADA Web* (tabela 10), representados pelo diagrama de pacotes, e os atores que intervêm nesta aplicação. O utilizador pode consultar o manual de ajuda da aplicação e terminar sessão na aplicação *Web* sempre que pretenda. As funcionalidades gerais representadas na figura 12 através de pacotes são:

- *Dashboard* - Apresenta os *links* para as diferentes funcionalidades a que o ator tenha acesso mediante as suas permissões;
- *Empresa* - Apresenta todas as informações da empresa;
- *Informa Pessoal* - Apresenta todas a informações referentes ao ator logado na aplicação;
- *Rede de Sensores* - Apresenta todas as informações referentes às várias redes de sensores que o ator tem configurado na aplicação *Web*;
- *Alarme* - Apresenta a lista de alarmes ocorridos nas redes de sensores;

- *Reporte* - Apresenta toda a informação sobre as redes de sensores através de gráficos e tabelas e permite ao ator pesquisar várias informações sobre as redes de sensores;
- *Suporte* - Permite aos atores colocar dúvidas sobre a aplicação *Web* ao Administrador da aplicação;
- *Configuração* - Permite fazer todas as configurações necessárias às redes de sensores na aplicação *Web*;
- *Administração* - Permite fazer todas as configurações necessárias para as empresas clientes da *eneidaws* poderem ter acesso à aplicação *Web*.

No Anexo A, apresentam-se os diagramas de casos de uso complementares para cada um dos pacotes vistos na figura 12.

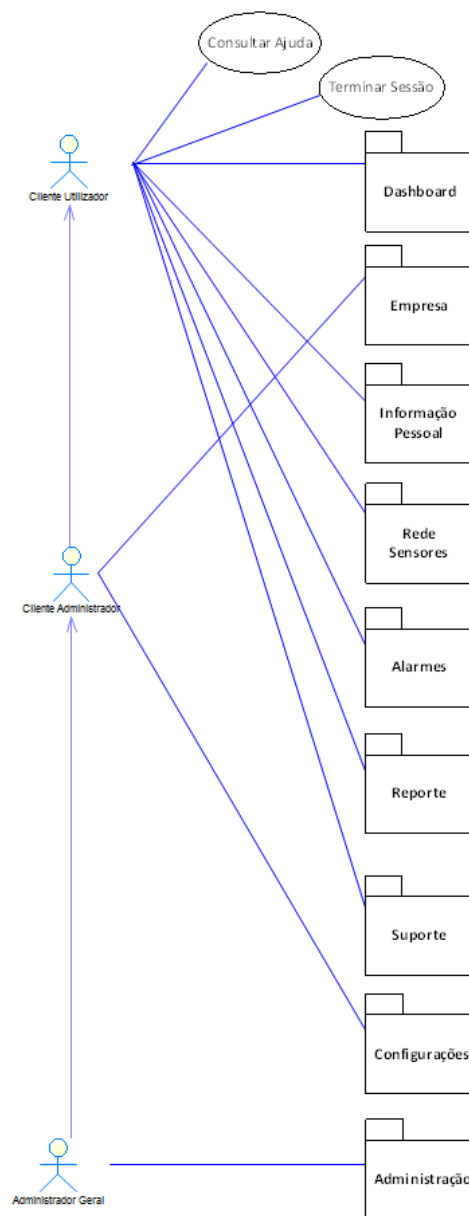


Figura 12 - Diagrama de Pacotes e Casos de Uso das Funcionalidades Gerais Aplicação Web SCADA

4.7.3. Requisitos Não Funcionais

Os requisitos não funcionais, considerados para a aplicação *SCADA Web*, são os seguintes, como se pode ver na tabela 11.

Tabela 11 - Requisitos não funcionais aplicação SCADA Web

Código ²⁵	Requisitos Não Funcionais
RNF1ASW	Usabilidade – A aplicação deve ter um <i>interface</i> simples e <i>user friendly</i> capaz de funcionar em diferentes dispositivos através do <i>framework Bootstrap</i> ²⁶ .
RNF2ASW	Performance – A representação de gráficos e tabelas com os últimos valores lidos numa rede não deve ser superior a 60 segundos.
RNF3ASW	Escalabilidade – A aplicação Web deve ser capaz de suportar vários utilizadores a aceder à plataforma em simultâneo.
RNF4ASW	Disponibilidade – A aplicação deve ter capacidade de adaptação quando um requisito muda e estar disponível para o utilizador de forma permanente.
RNF5ASW	A aplicação SCADA web deve ser alojada em sistemas operativos Windows no servidor <i>Internet Information Server (IIS)</i> .
RNF6ASW	O idioma da aplicação deve ser o inglês.
RNF7ASW	A aplicação deve ter uma arquitetura de desenvolvimento modular.
RNF8ASW	Deve ser possível configurar em <i>eXtensible Markup Language (XML)</i> a <i>string</i> de conexão à base de dados <i>SQL Server</i> .

4.8. VANTAGENS PREVISTAS DO NOVO SISTEMA

A elaboração deste projeto, para o novo sistema *SCADA* será composta por quatro aplicações, as quais trarão à *eneidaws* as seguintes vantagens:

²⁵ **RNF1ASW** – Requisito Não Funcional número 1, Aplicação **SCADA Web**

²⁶ O *Bootstrap* é o *framework* criado por Mark Otto e Jacob Thornton, ambos engenheiros do *Twitter*; trata-se de uma coleção de componentes para criação de *websites* e aplicações *web* utilizando o *HyperText Markup Language, version 5 (HTML5)* e *Cascading Style Sheets (CSS)*.

- **Vantagens da Estrutura em Módulos**

- Cada uma das quatro aplicações a serem desenvolvidas representa um módulo para o novo sistema *SCADA*.
- A estrutura do novo sistema *SCADA*, ao ser definida por módulos, permite maior flexibilidade na utilização de cada um relativamente às configurações das soluções que *eneidaws* queira apresentar aos seus clientes;
- O desenvolvimento de cada uma das aplicações apresentará uma arquitetura organizada e segmentada, que tornará as aplicações mais flexíveis para implementação de funcionalidades atuais e futuras;
- O facto de as aplicações serem definidas por módulos permite que as mesmas possam ser removidas do sistema *SCADA*, admitindo a introdução de novas aplicações/módulos que possam ser desenvolvidas futuramente ou simplesmente a adição de novas aplicações/módulos.

- **Vantagens Aplicação Servidor**

- Ao manter a mesma estrutura no formato das mensagens, garante-se que as comunicações do novo sistema funcionarão com os atuais equipamentos da *eneidaws* e respetivas redes de sensores;
- O servidor apresenta a opção que permite ao utilizador ativar ou desativar a análise de alarmes. Caso esta opção esteja ativa, o servidor tem a capacidade de verificar alarmes em todas as mensagens recebidas, permitindo notificar instantaneamente o utilizador por *SMS* e *e-mail*.
- No sistema atual da *eneidaws*, só a rede de sensores tem capacidade de notificar o utilizador por *SMS*, caso existam alarmes. Este *software*, ao ter capacidade de notificar o utilizador com a existência de alarmes por *e-mail* e *SMS*, permitirá redundância na notificação de alarmes, algo fulcral nos sistemas críticos;
- O servidor será capaz de funcionar em modo partilhado com várias redes de sensores ou funcionar em modo dedicado com uma única rede. Esta configuração será sempre escolhida de acordo com o que a *eneidaws* pretenda;

- O servidor terá a capacidade de armazenar todos os dados recebidos das diferentes redes de sensores dos seus clientes, guardando os dados nas bases de dados únicas de cada cliente. Desta forma permite-se a portabilidade dos dados do cliente de uma forma fácil e segura.

- **Vantagens Aplicação Serviço de Alarmes**

- Esta aplicação funciona exclusivamente para a verificação da existência de alarmes nas redes de sensores e notifica instantaneamente o utilizador por *e-mail* ou *SMS*;
- Caso a *eneidaws* o pretenda, pode ter vários serviços de alarme em execução.

- **Vantagens Aplicação Web Service**

- A *eneidaws*, ao ter um *Web Service*, permitirá acesso à informação através de pedidos *HTTP* relativamente a uma rede de sensores; permite, desta forma, a fácil integração com outros sistemas, sendo uma grande vantagem no uso para sistemas distribuídos;
- A aplicação *Web Service* está preparada para suportar a receção de informação através de pedido *HTTP* por *POST*, guardando os dados na base de dados. No futuro, se a *eneidaws* pretender, pode colocar o seu *hardware* a fazer *POST* de dados da rede de sensores para *Web Service*, sem necessitar de utilizar a aplicação servidor;
- A existência desta aplicação permite, no futuro, que a *eneidaws* desenvolva aplicações nativas para outros sistemas operativos móveis. Estas aplicações terão acesso aos dados, consumindo informação do *Web Service*;
- Existe ainda a possibilidade de disponibilizar informação das redes de sensores aos clientes que não pretendam acesso à aplicação *SCADA Web* e que queiram somente os dados adquiridos dos equipamentos da *eneidaws*.

- **Vantagens Aplicação SCADA Web**

- Com uma arquitetura modular e flexível, tornar-se-á mais fácil a implementação de novas funcionalidades;

- Esta aplicação, ao ser desenvolvida numa vertente *Web*, permite que a mesma seja multiplataforma, funcionando em diferentes sistemas operativos e diferentes dispositivos, bastando para isso o recurso a um *browser*;
- Sendo uma aplicação *Web*, permite que a informação esteja centralizada e facilita as atualizações e correções de erros para todos os clientes, não existindo obrigatoriedade na instalação da aplicação em cada computador que necessite aceder a esta;
- A integração de controlo de acessos é uma grande mais-valia, dado que na aplicação anterior não existe qualquer controlo sobre o que determinado utilizador pode fazer na aplicação. Esta funcionalidade permite saber o que um utilizador fez em toda a aplicação, uma vez que se interage com sistemas críticos, será importante ser criterioso na atribuição de responsabilidades;
- A existência de controlo de acessos permite, de acordo com diferentes níveis de privilégios, disponibilizar de forma dinâmica o acesso aos conteúdos da aplicação *Web*, por cada utilizador;
- Os conteúdos da aplicação *Web* são disponibilizados de forma dinâmica. Cada utilizador tem atribuídos diferentes privilégios de acesso à aplicação e, com base nos mesmos, as funcionalidades ficam disponíveis de acordo com o privilégio do utilizador.
- O facto de se tratar de uma aplicação *online* permite que a mesma esteja disponível de forma independente do género de equipamento;
- A *eneidaws* terá o administrador geral, que tem como funções: criar e configurar licenças; associar licenças a uma empresa; criar administradores associados a essa empresa; e prestar suporte aos clientes da aplicação *Web*. A licença permite que determinada empresa tenha ou não acesso à função *reporte* (análise de dados gráficos, tabelas, pesquisa de informação, etc.), dependendo da licença atribuída;
- Esta aplicação permite ao cliente da *eneidaws* ter acesso às seguintes opções: criar os seus próprios utilizadores e definir os respetivos níveis de privilégio; criar regras para alarmes e definir que utilizadores os recebem; criar redes de sensores e importá-las ou exportá-las automaticamente; visualizar os últimos dados adquiridos por uma rede de sensores; analisar histórico de alarmes e histórico de dados adquiridos em gráficos ou tabelas; exportar esta informação para *Excel* ou *PDF*, entre outros formatos; pesquisar por determinada informação de uma rede e, em caso de dúvida, usar suporte para contactar o administrador da plataforma. Em suma, a aplicação estará mais rica em funcionalidades e opções, tanto para os clientes como para a *eneidaws*.

4.9. PROTÓTIPOS DEFINIDOS PARA AS APLICAÇÕES

Para conceber os protótipos das aplicações, foram produzidos *mockups* para cada aplicação. Os *mockups* serviram para ilustrar graficamente as aplicações a nível de características e funcionalidades, para a *eneidaws* poder avaliar se seria esse o resultado pretendido para cada uma das aplicações, antes do início da implementação das aplicações. No Anexo B, apresentam-se todos os *mockups* definidos para a prototipagem de cada uma das aplicações.

Neste capítulo apresentou-se a descrição da solução proposta para o novo sistema *SCADA*, os *mockups* gráficos definidos para os protótipos de cada uma das aplicações e todos os requisitos necessários para o desenvolvimento de cada aplicação e de todo o sistema *SCADA*. No capítulo 5 será apresentado o enquadramento da arquitetura teórica definida para cada aplicação, tendo em conta todos requisitos definidos no capítulo 4.

5. ENQUADRAMENTO TEÓRICO DA ARQUITETURA

Neste capítulo apresenta-se a arquitetura teórica definida para a elaboração deste projeto e, em particular, a arquitetura definida para cada uma das aplicações desenvolvidas, tendo em conta os requisitos vistos anteriormente.

Apresentam-se as vantagens e desvantagens da utilização destas arquiteturas em cada uma destas aplicações. No final mostram-se as ferramentas e tecnologias necessárias para implementar as quatro aplicações para o novo sistema e o novo modelo de dados.

5.1. DESCRIÇÃO GERAL DO SISTEMA

Na figura 13, apresenta-se a visão geral do sistema *e-Diagnostic Software* composto por quatro aplicações para monitorização e controlo das redes de sensores. O novo sistema SCADA é composto por um servidor central com sistema operativo *Windows Server*, onde se encontra instalado o motor de base de dados *Microsoft SQL Server* para alojar as bases de dados do novo sistema SCADA, e o servidor *Web Internet Information Server (IIS)* para alojar a aplicação *SCADA Web* e a aplicação baseada em *Web Service*. No sistema central, encontram-se também as seguintes aplicações: a aplicação servidor executável, que recebe os dados das redes de sensores e armazena a informação em base de dados; a aplicação serviço de alarmes executável, responsável por verificar a existência de alarmes, fazendo periodicamente consulta de alarmes à base de dados; a aplicação *Web Service*, que permite disponibilizar a informação das redes de sensores via pedidos *HTTP* para integração com outros sistemas ou aplicações; por fim, a aplicação *SCADA Web*, que serve para supervisão e análise das redes de sensores via *Web* através de um *browser*. Todo este sistema está disponível pela internet, para os utilizadores poderem aceder à aplicação *SCADA Web* e à aplicação baseada em *Web Service*. A *gateway* da rede de sensores tem de ter acesso à internet para conseguir estabelecer uma ligação *TCP/IP* por *socket* com a aplicação servidor de modo a reportar todos os dados da rede de sensores.

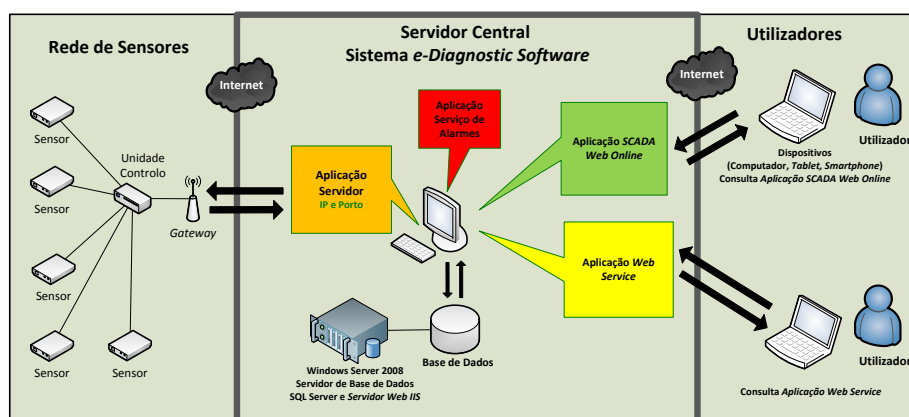


Figura 13 - Visão geral do sistema *e-Diagnostic software* para monitorização das redes de sensores

5.2. ARQUITETURA TEÓRICA DAS APLICAÇÕES

Nesta secção apresentam-se as arquiteturas definidas para cada uma das aplicações e para o modelo de dados, sendo explicadas as vantagens e desvantagens na utilização de cada uma destas. Como se pretende que estas aplicações sejam modelares e flexíveis para integração das atuais e futuras funcionalidades, foram usadas diferentes arquiteturas. As arquiteturas escolhidas tiveram em conta os requisitos pretendidos para implementação de cada uma das aplicações. Na secção 5.3 apresentam-se as ferramentas e tecnologias usadas para ser possível a implementação das quatro aplicações e do modelo de dados.

5.2.1. Estrutura da Base de Dados

Com base nos requisitos pretendidos pela empresa, foram criadas duas bases de dados capazes de suportar todo o sistema (informação mais detalhada sobre estes dois modelos de dados será apresentada no capítulo 6).

Este género de estrutura, pedido pela empresa com uso de duas bases de dados (BD), é inspirado no conceito da arquitetura *Multi-Tenant Data* (Microsoft, 2006), que prevê, por exemplo, que cada empresa é um *tenant* e cada um destes *tenants* representa uma base de dados física para cada empresa. Existe a necessidade de haver uma base de dados comum para todas as empresas, onde é feita a associação do *tenant* à empresa; sempre que uma empresa/*tenant* pretenda consultar os seus dados, é redirecionada para a base correspondente a esse *tenant*. Em concreto, na estrutura de dados definida para este projeto, existe uma BD central para validação das contas de *login* dos clientes da *eneidaws*, onde é feito o redirecionamento para a BD física deste cliente, sendo posteriormente apresentados os seus dados, como se pode ver na figura 14. Nesta figura, o utilizador Eduardo faz *login* na aplicação; após ser validado que este utilizador pertence à empresa *id* 1, é redirecionado para a base de dados das redes de sensores da sua empresa, para tratamento e consulta da sua informação.

Com base nesta estrutura definida para a base de dados do novo sistema *SCADA*, definiram-se as duas base de dados seguintes:

- Uma base de dados foi criada para guardar: todas as empresas criadas no sistema; licenças criadas e associadas empresa a empresa; contas de *login* criadas e associadas a uma empresa, contando com diferentes níveis de privilégios para acesso ao sistema.
- A segunda base de dados foi criada para guardar todos os dados adquiridos nas redes de sensores. Cada cliente que exista na aplicação *SCADA Web* tem a sua base de dados única com os seus dados, havendo desta forma uma separação física dos dados de cada cliente.

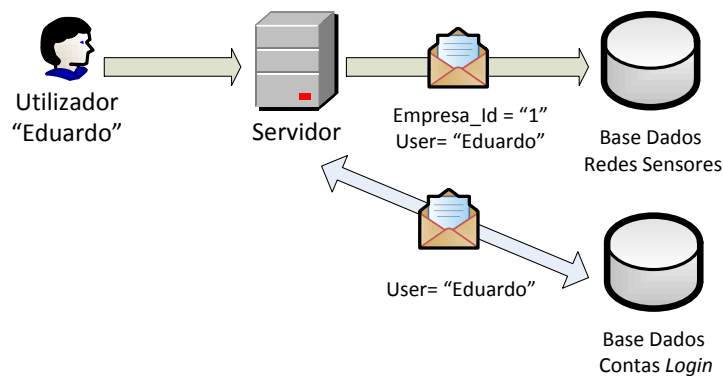


Figura 14 - Exemplo Estrutura de Dados baseada na arquitetura Multi-Tenant Data

Vantagens no uso desta estrutura de dados

- Portabilidade dos dados de cada cliente de uma forma fácil e segura;
- Os clientes que, numa fase inicial, queiram usar o sistema no ambiente *online*, e, mais tarde, pretendam ter todo o sistema dentro da sua empresa, têm a oportunidade de transportar unicamente a sua base de dados, sem conter dados de outros clientes;
- Clientes que tenham como restrição os seus dados não estarem envolvidos com os de outros clientes têm esta limitação ultrapassada;
- Sempre que o utilizador faça *login* na aplicação é automaticamente redirecionado para a sua base de dados das redes de sensores, sem que o mesmo se aperceba.

Desvantagens no uso desta estrutura de dados

- Na eventualidade de ser necessário adicionar uma nova funcionalidade na aplicação e que envolva mexer na estrutura da base de dados, será necessário atualizar, uma a uma, as bases de dados dos clientes.
- Com o recurso às duas bases de dados, pode tornar-se mais complexa a codificação das aplicações, sendo por isso necessário cuidado no desenvolvimento, de modo a haver uma boa separação dos dados e da lógica de negócio da aplicação.

Apesar das desvantagens na eventualidade de adição de novas funcionalidades e perante a necessidade de atualização das bases de dados de todos os clientes, é muito importante para a *eneidaws* ter a possibilidade de ter os dados de cada cliente separados, para portabilidade dos mesmos. Desta forma, optou-se por usar esta arquitetura para a estruturação da base de dados.

5.2.2. Arquitetura da Aplicação Servidor e da Aplicação Serviço de Alarmes

A arquitetura que se considerou mais adequada para o desenvolvimento das aplicações Servidor e Serviço de Alarmes, tendo em conta a análise de requisitos, foi a estruturação em “N Camadas” (Microsoft, 2013), vista na figura 15, desenvolvida com tecnologia *Windows Forms*.

- **Camada de Apresentação:** A camada de apresentação possui a responsabilidade de apresentar os dados resultantes das operações. Consiste na *interface* gráfica da aplicação, que torna possível a interação com o utilizador. Esta camada trabalha com resultados recebidos pela camada de negócio.
- **Camada de Negócio:** O objetivo da camada de negócio é implementar a lógica da aplicação, expondo esta lógica para a camada de apresentação, através de *interfaces* bem definidas. Neste sentido, a camada de negócio coordena a aplicação, processa comandos, efetua operações lógicas, avaliações, cálculos e processa os dados entre a camada de apresentação e a camada de acesso a dados.
- **Camada de Acesso aos dados:** A camada de acesso a dados determina o acesso à base de dados, executando um conjunto de código *Structured Query Language (SQL)* ou *Language-Integrated Query (LINQ)* para manipulação da estrutura de dados. Nesta camada serão escritos métodos genéricos que têm capacidade de interagir com a base de dados, tais como inserir, remover, atualizar ou consultar dados.
- **Camada de Dados:** A camada de dados representa o motor da base de dados onde os dados são guardados fisicamente. Neste sentido, optou-se por utilizar o servidor de base de dados *Microsoft SQL Server* que consiste no armazenamento dos dados normalmente implementado através de um sistema independente.

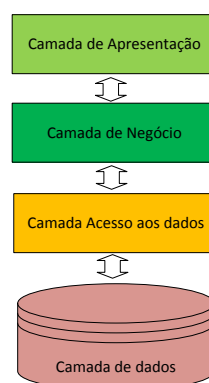


Figura 15 - Diagrama arquitetura "N camadas"

Vantagens no uso da Arquitetura “N Camadas”

- Separação muito clara entre as camadas de visualização e regras de negócios na aplicação servidor e na aplicação serviço de alarmes;
- A manutenção do sistema destas aplicações torna-se mais fácil por se encontrar tudo organizado em camadas;
- Centralização das camadas de negócio num componente, o que facilita a manutenção desse código, assim como o uso e a reutilização noutras aplicações que possam vir a ser feitas (por exemplo, novo servidor ou serviço de alarmes);
- Permite o desenvolvimento, testes e manutenção de forma isolada entre as camadas;
- O projeto destas aplicações passa a ter uma melhor organização na arquitetura de cada uma;
- Para a *eneidaws* torna o entendimento do projeto mais fácil para novos programadores que não participaram no desenvolvimento ou melhoramento destas aplicações;
- Torna estas aplicações escaláveis.

Desvantagens no uso da Arquitetura “N Camadas”

- Como esta arquitetura requer mais cuidados, o desenvolvimento é mais demorado numa fase inicial, dada a necessidade de criar componentes separados e definir bem a lógica para desenvolvimento aplicacional;
- Uma vez que o crescimento destas aplicações irá apresentar cada vez mais componentes para tratar, a lógica e o entendimento destas aplicações tenderá a ser mais complexo, para novos intervenientes num projeto com este género de arquitetura. No entanto, ter-se todas as camadas bem segmentadas facilita a trabalho com esta arquitetura;
- Exige muita disciplina dos programadores relativamente à separação das camadas, o que gera alguma complexidade aos programadores numa fase inicial.

Tendo em conta as vantagens e desvantagens encontradas nesta arquitetura, considerou-se que as vantagens seriam mais relevantes que as desvantagens, tendo em conta que as aplicações desenvolvidas ficam com uma arquitetura bem estruturada, facilitando a adição de funcionalidades futuras simultaneamente, torna-se mais fácil fazer manutenção e por estas serem escaláveis. Desta forma, optou-se por usar esta arquitetura para a estruturação da aplicação servidor e da aplicação serviço de alarmes.

5.2.3. Arquitetura da Aplicação Web Service

A arquitetura que se considerou mais adequada para o desenvolvimento da aplicação *Web Service*, tendo em conta a análise de requisitos, foi a arquitetura *Windows ASP.NET Web API*. Pretendia-se um serviço *Web* capaz de disponibilizar o acesso à informação das redes de sensores e possibilita-se a integração com outros sistemas. Considerando que a tecnologia *Windows Web API* permite o desenvolvimento de serviços *HTTP* que podem ser “consumidos” por uma ampla gama de clientes, incluindo *browsers*, *smartphones*, *tablets*, esta era a arquitetura que mais se adequava. Na figura 16 demonstra-se o exemplo do funcionamento da arquitetura *WEB API*.

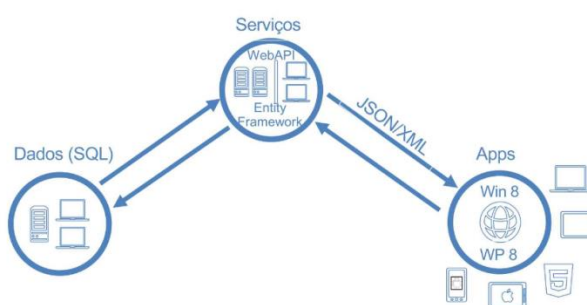


Figura 16 - Exemplo do funcionamento da Arquitetura Web API (TDC2013, 2013)

Vantagens no uso Web Services

- Independência da plataforma e linguagem de programação, permitindo que programas escritos em diferentes linguagens e diferentes plataformas comuniquem uns com os outros de uma forma *standard*; para este projeto, um exemplo disto será o desenvolvimento de aplicações *android*, *iphone*, entre outras que consumam informação das redes de sensores do *Web Service*;
- São flexíveis, escaláveis e reutilizáveis, o que permite que sejam registados de modo a que potenciais utilizadores possam localizá-los com maior facilidade, ou seja, os clientes de *eneidaws* terão acesso a vários métodos para consulta de informação através de um *link*, tendo cada utilizador uma chave única para fazer pedidos ao *Web Service*. Aplicações deste género são flexíveis para adição de novos métodos para consulta e fáceis de reutilizar;
- Reduzem custos, automatizando a interação entre aplicações e processos de negócio; diminuem os custos de transação e minimiza-se o erro humano. Além disso, os *web services* são componentes reutilizáveis cujas funcionalidades e conteúdos existentes podem ser facilmente melhorados, reduzindo assim o custo de desenvolvimento.

Desvantagens no uso *Web Services*

- Para utilização dos *Web Services* é necessário tempo de aprendizagem sobre este tipo de aplicação, sendo essencial compreender o funcionamento através de documentação associada ao mesmo, que a *eneidaws* fornecerá aos seus clientes;
- Este tipo de aplicação, normalmente, só é usado por especialistas na área das tecnologias de informação, não sendo fácil para o utilizador comum; só alguns clientes da *eneidaws* terão capacidade para usar este serviço *Web*.

Tendo em conta as necessidades da empresa em ter uma aplicação baseada em *Web Service*, esta arquitetura representa integralmente a facilidade na adição de novos pedidos e na disponibilização de dados das redes de sensores para serem consumidos por outras aplicações externas ao novo sistema *SCADA*. Desta forma, optou-se por usar esta arquitetura para a estruturação da aplicação *Web Service*.

5.2.4. Arquitetura da Aplicação *SCADA Web*

A arquitetura que se considerou mais adequada para o desenvolvimento da aplicação *SCADA Web*, tendo em conta a análise de requisitos, foi a arquitetura *Model-View-Controller (MVC)*, com a tecnologia *ASP.NET MVC* (Wikipédia, 2014a). Na figura 17, apresenta-se o exemplo do funcionamento da arquitetura *MVC*. Esta arquitetura permite a divisão da aplicação em várias camadas muito bem definidas (Protocolo TI, 2012), as quais se subdividem em três:

- **Model:** O modelo é utilizado para manipular informações de forma mais detalhada, sendo recomendada, sempre que possível, a utilização dos modelos para realizar consultas, cálculos e todas as regras de negócio do *site* ou sistema. É o modelo que tem acesso a toda e qualquer informação vinda da base de dados.
- **View:** A vista apresenta todo o conteúdo de um modelo. Recebe os dados do modelo e especifica os dados que devem ser apresentados. Atualiza também a apresentação de dados quando existem mudanças no modelo. A vista permite o contacto do utilizador com o controlador.
- **Controller:** O controlador define o comportamento da aplicação. Trata os pedidos do utilizador e indica à vista o que deve ser apresentado. Além disso, interpreta os pedidos do utilizador e mapeia-os em ações a serem executadas pelo modelo. Os pedidos dos utilizadores são feitos por protocolo *HTTP* através de *GET* e *POST*.

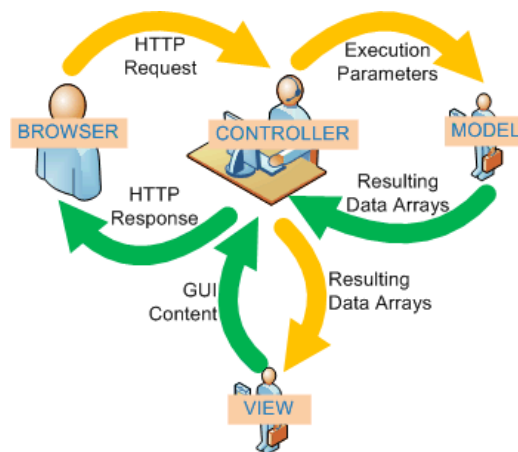


Figura 17 - Exemplo do Funcionamento da Arquitetura MVC (Feng Hua, 2014)

Vantagens no uso da arquitetura MVC

- Separação muito clara entre as camadas de visualização e regras de negócios, pois cada funcionalidade na aplicação servidor corresponde a um controlador;
- A manutenção do sistema torna-se mais fácil por esta aplicação ter várias camadas;
- Reaproveitamento de código, podendo ser reutilizado em outros projetos da *eneidaws*, ou para construção de novos controladores;
- É possível ter desenvolvimento em paralelo do modelo, da vista e do controlo dado estes serem independentes, pela equipa de desenvolvimento da *eneidaws*;
- As alterações na camada de visualização não afetam as regras de negócios já implementadas na camada do modelo, permitindo à *eneidaws* modificar o aspeto gráfico da aplicação *Web*, sempre que pretenda;
- Permite o desenvolvimento, testes e manutenção de forma isolada entre as camadas e em cada controlador desenvolvido;
- O projeto passa a ter uma melhor organização na sua arquitetura;
- Para a *eneidaws*, torna o entendimento do projeto mais fácil a novos programadores que não participaram no desenvolvimento ou melhoramento destas aplicações;
- Torna a aplicação escalável.

Desvantagens no uso da arquitetura MVC

- Em sistemas de baixa complexidade que possam existir na *eneidaws*, o MVC pode criar mais complexidade desnecessária no desenvolvimento;

- Uma vez que o crescimento destas aplicações irá apresentar cada vez mais componentes para tratar, a lógica e o entendimento destas aplicações tenderá a ser mais complexo, para novos intervenientes num projeto com este género de arquitetura. No entanto, ter-se toda a estrutura da aplicação bem segmentada facilita o trabalho com esta arquitetura;
- Exige muita disciplina dos programadores relativamente à separação das camadas, o que gera alguma complexidade aos programadores numa fase inicial.

Tendo em conta as vantagens e desvantagens encontradas, considerou-se que esta arquitetura perfaz as necessidades pretendidas, por existir uma estrutura lógica na organização e desenvolvimento da aplicação *Web*, que permitirá a integração de novos controladores com funcionalidades futuras para a aplicação *Web*, tornando a aplicação escalável. Desta forma, optou-se por usar esta arquitetura para a estruturação da aplicação *SCADA Web*.

5.3. FERRAMENTAS E TECNOLOGIAS UTILIZADAS

As ferramentas e tecnologias usadas que mais se adequaram ao desenvolvimento das aplicações neste projeto, tendo em conta as arquiteturas teórica descritas e a descrição dos requisitos propostos, foram as seguintes:

Visual Studio 2012 – O *Microsoft Visual Studio* é um pacote de programas da *Microsoft* para desenvolvimento de *software* especialmente dedicado ao *.NET Framework* e às linguagens *Visual Basic (VB)*, *C*, *C++*, *C# (C Sharp)* e *J# (J Sharp)*. É também um ótimo produto de desenvolvimento para ambiente *web*, com recurso à plataforma *ASP.NET*. Esta ferramenta foi escolhida para desenvolvimento das quatro aplicações por ser a ferramenta adequada para trabalhar em tecnologias *Windows* e pelo facto de a empresa usar esta ferramenta no desenvolvimento das suas aplicações.

Microsoft SQL Server 2012 – O *Microsoft SQL Server* é um sistema de gestão de base de dados (SGDB) relacional desenvolvido pela *Microsoft*, onde se podem criar bases de dados para armazenamento e consulta de informação. Este servidor de base de dados foi escolhido por ser o que a empresa usa atualmente e por ser compatível com as arquiteturas a implementar.

Microsoft SQL Management Studio – O *SQL Server Management Studio* é um *software* que serve para configurar, gerir e administrar todos os componentes dentro do motor de base de

dados *Microsoft SQL Server*. Esta ferramenta inclui o editor de *scripts* e ferramentas gráficas que trabalham com objetos e recursos do servidor. O uso desta ferramenta serviu para a configuração das bases de dados, contas de utilizadores, permissões, sistema de *backups* e para consulta dos dados durante o desenvolvimento das aplicações.

VisualSVN – *VisualSVN* é um cliente que pode ser instalado e integrado no *Microsoft Visual Studio*, fornecendo uma *interface* para executar as operações de controlo de versão mais comuns diretamente dentro do *IDE Visual Studio*. O uso desta aplicação serviu para criar várias versões no desenvolvimento das aplicações e, sempre que surgisse um problema, podia fazer-se a recuperação de versões anteriores. Por outro lado, serviu para a organização do trabalho feito e na progressão dos trabalhos.

Fiddler 2 – O *Fiddler* é uma aplicação para captura de tráfego *HTTP/HTTPS* para fins de resolução de problemas. Esta ferramenta foi importante para efetuar testes de pedidos *GET* e *POST* em formato *JSON*, realizados à aplicação *Web Service*, e para executar testes com pedidos *GET* e *POST* realizados aos controladores da aplicação *SCADA Web*. Em particular muitos pedidos realizados funcionam de forma assíncrona, não sendo fácil a deteção do problema sem ajuda desta ferramenta.

Talend Open Studio 5.5 – O *Talend Open Studio* é um *software open source* que permite a integração de dados, gestão de dados, entre outros. Neste caso em particular, esta ferramenta foi importante para a migração dos dados das redes de sensores da base de dados atual da *eneidaws* para a nova estrutura de base de dados implementada para este projeto.

Microsoft Office 2010 – É uma ferramenta que contempla um conjunto de aplicações, tal como *Word*, *Excel*, *Visio*, *PowerPoint*. O uso desta ferramenta foi importante para a produção de diagramas, desenvolvimento documental do relatório e para a elaboração dos *slides* da apresentação deste projeto.

Paint.Net – O *Paint.NET* é uma aplicação simples para edição e tratamento de imagens. Esta aplicação foi usada para criar e editar imagens e ícones, para as aplicações desenvolvidas.

Balsamiq Mockups – Esta ferramenta serve para criar protótipos, normalmente antes do desenvolvimento das aplicações. Em concreto para este projeto, após o levantamento dos requisitos foram produzidos alguns *mockups* para os protótipos das aplicações antes do desenvolvimento, para clarificar o aspeto que as aplicações iriam ter.

PowerDesigner 16.5 – É um *software* produzido pela empresa americana Sybase, que permite aos utilizadores suportar algumas fases e tarefas no processo de desenvolvimento de *software* ou sistemas de informação. Esta ferramenta serviu para criar os modelos entidade relacionamento e os modelos físicos das bases de dados.

Bootstrap – O *Bootstrap* é uma coleção de vários elementos e funções personalizáveis (botões, animações, menus, formulários, etc.), para projetos *web*, incluídos previamente numa única ferramenta. Ao projetar um *site* com o *Bootstrap*, pode-se escolher quais os elementos que se quer usar. O mais importante é ter a certeza de que os elementos escolhidos não entram em conflito entre si. Estes elementos personalizáveis contidos no *Bootstrap* são uma combinação de *HTML5*, *Cascading Style Sheets (CSS)* e várias opções de *plug-ins jQuery*. O *template* escolhido para aplicação *SCADA Web* tem como base esta *framework*, que está preparada para funcionar em diferentes dispositivos; sendo o *template responsive*, ajusta-se à área do dispositivo que abrir este *template*, e a nível gráfico tornou-se mais *user-friendly*.

Framework .Net – A *Framework .Net* é a base de todas as aplicações desenvolvidas em ambiente de tecnologias *Windows*. Tendo sido as quatro aplicações desenvolvidas em tecnologias *Windows*, neste projeto recorreu-se à *Framework .Net 4.5*. Esta consiste em dois componentes principais, ou seja, é executada sobre uma *Common Language Runtime (CLR)* (Ambiente de Execução Independente de Linguagem), interagindo com a *Framework Class Library (FCL)* (Conjunto de Bibliotecas Unificadas). A *CLR* permite a execução de diferentes linguagens de programação, interagindo entre si como se fossem uma única linguagem.

ASP.NET MVC – O *ASP.NET MVC* é uma arquitetura para desenvolvimento de aplicações *Web* baseadas em *MVC*, sendo esta *framework MVC* definida pelo *namespace System.Web.Mvc*, assente na *framework ASP.NET* e esta por sua vez na *framework .NET*. Para o desenvolvimento deste projeto utilizou-se o *ASP.NET MVC 4* e o *Razor view engine* para criar as várias vistas da aplicação, conjuntamente com *HTML* e *CSS*.

WEB API – A *WEB API* é uma tecnologia da *Microsoft* para o *ASP.NET framework*, que serve para o desenvolvimento de serviços *HTTP* para disponibilização de informação a outras aplicações, normalmente expressas em *JSON* ou *XML*. Esta foi usada para o desenvolvimento da aplicação baseada em *Web Service*, que permite disponibilização de informação das redes de sensores e submissão de dados no formato *JSON*.

C# – Esta foi a linguagem de programação que mais se adequou ao desenvolvimento das quatro aplicações nas diferentes tecnologias, e também por ser usada no desenvolvimento de aplicações na *eneidaws*. O **C#** é uma linguagem de programação orientada a objetos desenvolvida pela *Microsoft* como parte integrante da plataforma *.NET*. A sua sintaxe orientada a objetos foi baseada no **C++**, mas inclui muitas influências de outras linguagens de programação, como *Delphi* e *Java*.

Entity Framework – Os modelos existentes nas aplicações desenvolvidas neste projeto usam o *Entity Framework* para fazer o mapeamento das entidades na base de dados para objetos. O *Microsoft ADO.NET Entity Framework* é uma *framework* do tipo *Object Relational Mapping (ORM)* que permite trabalhar com dados relacionais, como objetos de domínio específico, eliminando a necessidade da maior parte dos códigos de acesso aos dados que geralmente é preciso devolver.

LINQ – Visto usar-se neste projeto *Entity Framework*, é necessária uma linguagem específica que permita a consulta e manipulação dos dados. A *Language Integrated Query (LINQ)* é um componente do *Microsoft .NET* que adiciona funcionalidades de consulta em algumas linguagens de programação em *.NET*. O **LINQ** corresponde a uma sintaxe unificada, inicialmente incorporada nas linguagens **C#** e *Visual Basic*, para consultas em fontes de dados variadas. A sintaxe de consulta da **LINQ** foi inspirada na da *Structured Query Language (SQL)*, que é uma linguagem padrão para comunicação com base de dados relacionais.

SQL – A *Structured Query Language (SQL)* é uma linguagem de pesquisa declarativa para base de dados relacional. No desenvolvimento deste projeto não existiu codificação com a linguagem **SQL**. No entanto, durante o desenvolvimento, foram feitas várias consultas à base de dados através do *Microsoft SQL Management Studio*.

jQuery – O *jQuery* é uma *framework* de *Javascript*, constituída por uma coleção de funções e métodos prontos para serem utilizados (por exemplo, validação de formulários, animações gráficas, etc.). O uso desta *framework* na aplicação *SCADA Web* teve como função principal a validação de dados do lado do cliente no *browser* (por exemplo, formulários); por outro lado, foi usada por ser compatível com a *framework* gráfica *Bootstrap*.

Ajax – O *Asynchronous Javascript and XML* é usado nos *browsers* para tornar as páginas *Web* mais interativas com os utilizadores, usando pedidos assíncronos de informações ao servidor *Web*. Uma das grandes vantagens do uso do *Ajax* na aplicação *SCADA Web* é existir a

capacidade de apresentar uma informação nova numa página/vista sem ser feito *refresh* à página onde foi solicitada uma informação nova.

JSON – O *JavaScript Object Notation* é um formato padrão leve e aberto que usa texto legível para transmitir dados em objetos que consistem em pares atributo valor. É utilizado principalmente para transmitir dados entre uma aplicação servidor *web*, como uma alternativa ao *XML*. Neste projeto recorreu-se ao formato *JSON* para disponibilização dos dados das redes de sensores através da nossa aplicação *Web Service* para futura integração dos dados.

Highcharts – O *Highcharts* é uma biblioteca para representar gráficos, escrita em *JavaScript*, oferecendo uma maneira fácil de adicionar gráficos interativos numa aplicação *Web*. Para integração com este projeto foi usado o *Highcharts .Net*, que é uma *Dynamic-link library (DLL)*, específica para integração com projetos *.NET*. O uso desta biblioteca serviu para desenhar e representar o histórico de dados das redes de sensores.

Google Maps – O *Google Maps* é um serviço de georreferenciação propriedade da Google. Recorreu-se ao uso desta *API* para representar as redes de sensores através das coordenadas *GPS* sendo estas configuradas e podendo ser visualizadas no mapa do *Google Maps*.

Google Webmail – O *Google WebMail* é o serviço de *e-mail* da Google. Recorreu-se ao uso desta *API* para envio de *e-mails* quando existem alarmes nas redes de sensores, para notificar o utilizador, e na área de suporte da aplicação *SCADA Web*.

BulkSMS – O *BulkSMS* é um serviço para envio de *SMS*, capaz de funcionar em diferentes linguagens de programação. Recorreu-se ao uso desta *API* para fazer o envio de *SMS* quando existem alarmes nas redes de sensores, para notificação dos utilizadores.

Neste capítulo apresentou-se a arquitetura teórica para cada uma das aplicações, sendo expostas as vantagens e desvantagens das mesmas. Por fim, apresentaram-se as ferramentas e tecnologias necessárias para implementação das aplicações neste projeto. No capítulo 6 apresenta-se os modelos implementados para gestão de dados de todo o sistema, e no capítulo 7 apresenta-se, de uma forma geral, a estrutura das aplicações com base nas arquiteturas referidas neste capítulo.

6. GESTÃO DE DADOS

Neste capítulo demonstra-se de uma forma simplificada os modelos de base de dados que mais se adequaram, tendo em conta a lógica de negócio definida para o sistema e para as aplicações.

6.1. MODELO DE DADOS

O modelo de base de dados que mais se adequou, tendo em conta a análise de requisitos e a lógica de negócio definida para o sistema, foi a modelação de duas bases de dados para suportar todas as necessidades do sistema.

A modelação da primeira base de dados (base de dados geral), que representa a realidade, foi definida tendo em conta as seguintes informações: a *eneidaws* tem os seus clientes que são empresas e cada empresa pertence a um país; as empresas cliente da *eneidaws* têm uma estrutura hierárquica que representa normalmente o chefe e os funcionários da empresa. Esta lógica foi transmitida para as entidades da base de dados, espelhando que cada empresa teria necessidade de várias contas de acesso à aplicação *SCADA Web*, com diferentes níveis de privilégios. Desta forma, seria possível representar na aplicação *SCADA Web* os utilizadores, com acesso a determinadas informações com base em regras de acesso. Do ponto de vista de negócio da *eneidaws*, existia o interesse em que cada empresa tivesse uma licença de acesso à aplicação *SCADA Web* e, mediante este género de licença, a empresa teria acesso a mais ou menos funcionalidades e opções na aplicação. Na secção 6.1.1, são apresentadas as entidades definidas para esta base de dados. Na secção 6.1.2 são apresentados os relacionamentos existentes entre estas entidades.

A modelação da segunda base de dados (Base de Dados Rede Sensores dos Clientes), que representa a realidade, foi definida tendo em conta as seguintes informações: cada empresa cliente da *eneidaws* tem redes de sensores; estas empresas têm as redes de sensores instaladas numa determinada localização, que pode representar uma zona numa fábrica ou uma cidade, etc; as redes de uma empresa com uma determinada localização são constituídas por vários sensores, sendo que cada sensor pode ser de um determinado tipo e modelo; os sensores de um determinado modelo podem ser agrupados para obterem determinadas informações (por exemplo, um grupo de sensores pode representar o meio ambiente numa empresa e, para isso, são usados sensores que adquiram temperatura e humidade ambiente); por fim, cada sensor tem entradas que são capazes de adquirir diferentes tipos de dados. Com base nos dados adquiridos nas entradas dos sensores, são definidas regras que representam alarmes sempre que as regras definidas sejam quebradas, prevenindo problemas existentes nas redes de sensores. Na secção 6.1.3, são apresentadas as entidades definidas para esta base de dados. Na secção 6.1.4, são apresentados os relacionamentos existentes entre estas entidades.

Estas bases de dados são centrais a todas as aplicações, conseguindo-se deste modo ter toda a informação num único ponto para posterior consulta e tratamento desses dados. Por outro lado, o motor de base de dados usado para suportar as bases de dados das aplicações é o *Microsoft SQL Server Express* gratuito ou versão completa, dependendo do contexto em que a *eneidaws* esteja a implementar o sistema.

Será igualmente importante configurar o motor de base de dados para funcionar em rede, para que as aplicações através de configurações por *XML* possam configurar a *string* de conexão ao motor de base de dados. Um outro elemento tido em conta foram os *backups* das bases de dados. Para a versão paga do *SQL Server* pode-se configurar *SQL Server Agent* para desempenhar esta tarefa. Para as versões *SQL Server Express* implementou-se um *Script* para fazer *backups* através de uma tarefa criada no sistema operativo, que chama o *Script* com a periodicidade que seja definida e efetua o *backup*.

6.1.1. Entidades Base de Dados Geral

Esta base de dados foi criada para guardar : todas as empresas criadas no sistema; licenças criadas e associadas empresa a empresa; contas de *login* criadas e associadas a uma empresa, segundo as quais estas têm diferentes privilégios de acesso ao sistema. Para a construção desta base de dados usou-se a *API .NET* para a vertente *web* a biblioteca *Simple Membership*, a qual permitiu a criação de contas de utilizadores com diferentes privilégios com recurso a regras, conseguindo-se, deste modo, garantir a segurança na aplicação. Por outro lado, tornou-se mais fácil a integração das nossas tabelas adicionais conjuntamente com as tabelas da biblioteca *Simple Membership*.

Nesta secção, descrevem-se as entidades principais referentes ao modelo de dados Base de Dados Geral.

- **Entidade Licenca** – A entidade “Licenca”, representa a informação relativa às licenças criadas e atribuídas a cada empresa criada no sistema.
- **Entidade Países** – A entidade “Países”, representa a informação relativa ao país de uma empresa criada no sistema.
- **Entidade Empresa** – A entidade “Empresa”, representa a informação relativa aos dados da empresa criada no sistema.
- **Entidade UserProfile** – A entidade “UserProfile”, referente à base de dados da *API .NET Simple Membership*, integrada na base de dados geral, representa a informação relativa aos utilizadores e contas de acesso, existentes no sistema associados a uma empresa.

- **Entidade *webpages_Roles*** – A entidade “*webpages_Roles*”, referente à base de dados da *API .NET Simple Membership*, integrada na base de dados geral, representa a informação relativa às regras de acesso e níveis de privilégios de cada utilizador referente ao sistema.
- **Entidade *webpages_Membership*** – A entidade “*UserProfile*”, referente à base de dados da *API .NET Simple Membership*, integrada na base de dados geral, representa a informação complementar dos utilizadores existentes na entidade “*UserProfile*”, tal como: último *login* efetuado, data em que foi criada a conta de utilizador, entre outros.
- **Entidade *Contato*** – A entidade “Contato” representa a informação correspondente à troca de mensagens realizadas dentro do sistema.
- **Entidade *Empresa_IDRede_Servidor*** – A entidade “*Empresa_IDRede_Servidor*”, representa a associação existente entre o Id único de uma rede criada e a empresa correspondente que tutela essa rede. Esta entidade é fundamental para a aplicação servidor saber, quando receba a mensagem de uma rede com um Id único, onde guardar essa informação na base de dados Redes de Sensores dessa empresa.

6.1.2. Relacionamentos Base de Dados Geral

Nesta secção, descrever-se-á os relacionamentos entre as entidades referentes, ao modelo de dados, Base de Dados Geral. No Anexo B, pode ver-se o “modelo físico” e “modelo entidade relacionamento” deste modelo de dados e respetivos relacionamentos que serão descritos nesta secção.

- **Relacionamento *Empresa_Paises***
Este relacionamento, (tabela 12), pretende expressar a relação existente entre as Entidades Empresa e Países. O objetivo é expressar quais as Empresas que têm um determinado País sendo que um País pode ter uma ou mais empresas.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Uma Empresa contém sempre um País;
- Um País pode conter varias empresas.

Tabela 12 - Relacionamento Empresa_Paises

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Empresa	Não	N : 1	Sim	Paises
Observações				
<ul style="list-style-type: none"> • Uma empresa só pode ter um país. • Uma empresa tem obrigatoriamente um país. • Um país pode ter várias empresas. 				

- **Relacionamento Licenca_Empresa**

Este relacionamento, (tabela 13), pretende expressar a relação existente entre as Entidades Licença e Empresa. O objetivo é expressar que uma Empresa tem uma Licença e uma Licença pertence a uma só Empresa.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Uma Empresa contém sempre uma Licença;
- Uma Licença corresponde a uma Empresa.

Tabela 13 - Relacionamento Licenca_Empresa

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Empresa	Não	1 : 1	Sim	Licenca
Observações				
<ul style="list-style-type: none"> • Uma empresa só pode ter uma licença. • Uma empresa tem obrigatoriamente uma licença. 				

- **Relacionamento UserProfile_Empresa**

Este relacionamento, (tabela 14), pretende expressar a relação existente entre as Entidades *UserProfile* e Empresa. O objetivo é expressar que uma Empresa tem um ou mais *UserProfiles* e um *UserProfiles* pertence a uma só empresa.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Uma Empresa contém um ou mais *UserProfiles*;
- Um *UserProfile* corresponde a uma Empresa.

Tabela 14 - Relacionamento *UserProfile_Empresa*

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Empresa	Sim	1 : N	Não	UserProfile
Observações				
<ul style="list-style-type: none"> Um <i>UserProfile</i> <u>só pode ter uma</u> empresa. Um <i>UserProfile</i> tem <u>obrigatoriamente</u> uma empresa. Uma empresa pode ter <u>vários</u> <i>UserProfiles</i>. 				

- **Relacionamento *webpages_UserInRoles***

Este relacionamento, (tabela 15), pretende expressar a relação existente entre as Entidades *webpages_Roles* e *UserProfile*. O objetivo é expressar que um *UserProfile* tem um ou mais *webpages_Roles* e uma *webpages_Roles* tem um ou mais *UserProfiles*.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Um *UserProfile* contém um ou mais *webpages_Roles*;
- Um *webpages_Roles* corresponde a um ou mais *UserProfile*.

Tabela 15 - Relacionamento *webpages_UserInRoles*

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
<i>UserProfile</i>	Não	N : N	Não	<i>webpages_Roles</i>
Observações				
<ul style="list-style-type: none"> Um <i>UserProfile</i> pode ter <u>vários</u> <i>webpages_Roles</i>. Um <i>webpages_Roles</i> pode ter <u>vários</u> <i>UserProfiles</i>. 				

6.1.3. Entidades Base de Dados Rede Sensores dos Clientes

Esta base de dados foi criada para guardar todos os dados adquiridos nas redes de sensores com todas as especificidades das redes. Cada cliente que exista na aplicação *SCADA Web* tem a sua base de dados única com os seus dados.

Nesta secção, descrevem-se as entidades principais referentes, ao modelo de dados, Base de Dados Redes Sensores dos Clientes.

- **Entidade Empresa_data** – A entidade “Empresa_data”, representa a informação relativa aos dados da empresa criada no sistema.
- **Entidade Rede** – A entidade “Rede”, representa toda a informação que uma rede de sensores contém.
- **Entidade Localização** – A entidade “Localização”, representa a informação relativa a uma localização a que uma rede de sensores corresponde.
- **Entidade Grupo** – A entidade “Grupo”, representa a informação relativa ao grupo de sensores associados a uma rede.
- **Entidade Sensor** – A entidade “Sensor”, representa a informação relativa aos sensores associados a uma rede
- **Entidade Modelo Sensor** – A entidade “Modelo Sensor”, representa a informação relativa ao modelo de um sensor associado a uma rede.
- **Entidade Entrada** – A entidade “Entrada”, representa a informação relativa às entradas de um sensor associado a uma rede.
- **Entidade Log_MensagensServidor** – A entidade “Log_MensagensServidor”, representa a informação relativa, a todos os dados gerados por uma rede de sensores.
- **Entidade Log_MensagensServidor_temp** – A entidade “Log_MensagensServidor_temp”, representa a informação relativa aos últimos dados gerados por uma rede de sensores.
- **Entidade Log_MensagensErroServidor** – A entidade “Log_MensagensErroServidor”, representa a informação relativa, a todos os erros verificados pela aplicação servidor, referente a uma rede de sensores.
- **Entidade Log_MensagensAlarmeServidor** – A entidade “Log_MensagensAlarmeServidor”, representa a informação relativa a todos os alarmes gerados por uma rede de sensores.
- **Entidade Regras_Rede** – A entidade “Entidade Regras_Rede”, representa a informação relativa, a todas as regras para alarmes referente às redes de sensores.
- **Entidade Regras_Acao** – A entidade “Regras_Acao”, representa a informação relativa a todas as ações que são desencadeadas quando se verifica a ocorrência de uma determinada regra.
- **Entidade Envia_Mensagem_Rede** – A entidade “Envia_Mensagem_Rede”, representa a informação relativa, a todas as mensagens de dados e ações enviadas pela aplicação *online* e por cada ação desencadeada.

- **Entidade Empresa_Contatos_Alarme** – A entidade “Empresa_Contatos_Alarme”, representa a informação relativa, a todos os utilizadores da empresa autorizados a receber por *e-mail* e *SMS*, a ocorrência de um alarme numa rede de sensores.
- **Entidade Control** – A entidade “Control”, representa a informação relativa as configurações feitas por um utilizador na aplicação *online*, guardando as suas informações.

6.1.4. Relacionamentos Base de Dados Rede Sensores dos Clientes

Nesta secção, pretende-se descrever os relacionamentos entre as entidades referentes, ao modelo de dados, Base de Dados Redes Sensores dos Clientes. No Anexo B, pode ver-se o “modelo físico” e o “modelo entidade relacionamento” deste modelo de dados e respetivos relacionamentos que serão descritos nesta secção.

- **Relacionamento Empresa_data_Rede**

Este relacionamento, (tabela 16), pretende expressar a relação existente entre as Entidades Empresa_data e Rede. O objetivo é expressar que uma Empresa tem uma ou mais Redes e uma Rede pertence a uma só Empresa.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Uma Rede contém sempre uma Empresa_data;
- Uma Empresa_data pode conter varias Redes.

Tabela 16 - Relacionamento Empresa_data_Rede

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Empresa_data	Sim	1: N	Sim	Rede
Observações				
<ul style="list-style-type: none"> • Uma rede <u>só pode ter uma</u> empresa_data. • Uma rede tem <u>obrigatoriamente</u> uma empresa_data. • Um empresa_data pode ter <u>várias</u> redes. 				

- **Relacionamento Rede_Localizacao**

Este relacionamento, (tabela 17), pretende expressar a relação existente entre as Entidades Rede e Localização. O objetivo é expressar que uma Rede tem uma Localização e uma Localização tem uma ou mais Redes.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Uma Rede contém sempre uma Localização;
- Uma Localização pode conter varias Redes.

Tabela 17 - Relacionamento Rede_Localizacao

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Rede	Não	N: 1	Sim	Localização
Observações				
<ul style="list-style-type: none"> • Uma rede só pode ter uma localização. • Uma rede tem obrigatoriamente uma localização. • Um localização pode ter várias redes. 				

- **Relacionamento Sensor_Rede**

Este relacionamento, (tabela 18), pretende expressar a relação existente entre as Entidades Sensor e Rede. O objetivo é expressar que uma Rede tem uma ou mais Sensores e um Sensor pertence sempre a uma Rede.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Um Sensor contém sempre uma Rede;
- Uma Rede pode conter vários Sensores.

Tabela 18 - Relacionamento Sensor_Rede

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Rede	Sim	1:N	Não	Sensor
Observações				
<ul style="list-style-type: none"> • Um sensor só pode ter uma rede. • Um sensor tem obrigatoriamente uma rede. • Uma rede pode ter vários sensores. 				

- **Relacionamento Sensor_ModeloSensor**

Este relacionamento, (tabela 19), pretende expressar a relação existente entre as Entidades Sensor e ModeloSensor. O objetivo é expressar que um Sensor tem um determinado ModeloSensor e um ModeloSensor pode pertencer a um ou mais Sensores.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Um Sensor contém sempre um Modelosensor;
- Um Modelosensor pode conter vários Sensores.

Tabela 19 - Relacionamento Sensor_ModeloSensor

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Sensor	Não	N:1	Sim	ModeloSensor
Observações				
<ul style="list-style-type: none"> • Um sensor só pode ter um modelosensor. • Um sensor tem obrigatoriamente um modelosensor. • Um modelosensor pode ter vários sensores. 				

- **Relacionamento Sensor_Grupo**

Este relacionamento, (tabela 20), pretende expressar a relação existente entre as Entidades Sensor e Grupo. O objetivo é expressar que um Sensor tem um determinado Grupo e um Grupo pode conter um ou mais Sensores.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Um Sensor contém sempre um Grupo;
- Um Grupo pode conter vários Sensores.

Tabela 20 - Relacionamento Sensor_Grupo

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Sensor	Não	N:1	Sim	Grupo
Observações				
<ul style="list-style-type: none"> • Um sensor só pode ter um grupo. • Um sensor tem obrigatoriamente um grupo • Um grupo pode ter vários sensores. 				

- **Relacionamento Entrada_Sensor**

Este relacionamento, (tabela 21), pretende expressar a relação existente entre as Entidades Entrada e Sensor. O objetivo é expressar que um determinado Sensor tem uma ou mais Entradas e uma Entrada pertence a um só Sensor.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Um Sensor contém sempre uma Entrada;
- Uma Entrada pode conter vários Sensores.

Tabela 21 - Relacionamento Entrada_Sensor

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Sensor	Sim	1:N	Não	Entrada
Observações				
<ul style="list-style-type: none"> • Uma entrada só pode ter um sensor. • Uma entrada tem obrigatoriamente um sensor. • Um sensor pode ter várias entradas. 				

- **Relacionamento Entrada_LogMensagensServidor**

Este relacionamento, (tabela 22), pretende expressar a relação existente entre as Entidades Entrada e LogMensagensServidor. O objetivo é expressar que uma determinada Entrada tem várias LogMensagensServidor e uma LogMensagensServidor pertence a uma só Entrada. A explicação do relacionamento Entrada_LogMensagensServidor temp é igual a esta explicação do relacionamento Entrada_LogMensagensServidor.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Uma LogMensagensServidor contém sempre uma Entrada;
- Uma entrada pode conter várias LogMensagensServidor.

Tabela 22 - Relacionamento Entrada_LogMensagensServidor

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Entrada	Sim	1:N	Não	LogMensagensServidor
Observações				
<ul style="list-style-type: none"> • Uma LogMensagensServidor só pode ter uma entrada. • Uma LogMensagensServidor tem obrigatoriamente uma entrada. • Uma entrada pode ter várias LogMensagensServidor. 				

- **Relacionamento Entrada_LogMensagensAlarmesServidor**

Este relacionamento, (tabela 23), pretende expressar a relação existente entre as Entidades Entrada e LogMensagensAlarmesServidor. O objetivo é expressar que uma determinada Entrada tem várias LogMensagensAlarmesServidor e uma LogMensagensAlarmesServidor pertence a uma só Entrada.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Uma LogMensagensAlarmesServidor contém sempre uma Entrada;
- Uma Entrada pode conter várias LogMensagensAlarmesServidor.

Tabela 23 - Relacionamento Entrada_LogMensagensAlarmesServidor

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Entrada	Sim	1:N	Não	LogMensagensAlarmesServidor
Observações				
<ul style="list-style-type: none"> • Uma LogMensagensAlarmesServidor só pode ter uma entrada. • Uma LogMensagensAlarmesServidor tem obrigatoriamente uma entrada. • Uma entrada pode ter várias LogMensagensAlarmesServidor. 				

- **Relacionamento Regra_Rede_Acao**

Este relacionamento, (tabela 24), pretende expressar a relação existente entre as Entidades Regra_Rede e Regra_Acao. O objetivo é expressar que uma determinada Regra_Rede tem uma ou mais Regra_Acao e Regra_Acao tem sempre uma Regra_Rede.

Após uma análise do funcionamento do sistema, definiram-se as seguintes condições:

- Uma Regra_Acao contém sempre uma Regra_Rede;
- Uma Regra_Rede pode conter várias Regras_Acao.

Tabela 24 - Relacionamento Regras_Rede_Acao

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Regra_Rede	Sim	1:N	Não	Regra_Acao
Observações				
<ul style="list-style-type: none"> • Uma Regra_Acao só pode ter uma Regra_Rede. • Uma Regra_Acao tem obrigatoriamente uma Regra_Rede. • Uma Regra_Rede pode ter várias Regras_Acao. 				

Neste capítulo apresentou-se o modelo de dados definido para as duas bases de dados, tendo sido apresentadas as respetivas entidades e os relacionamentos entre as mesmas. Estas bases de dados foram elaboradas para suportar todas as necessidades do sistema e o respetivo armazenamento de toda a informação referente às aplicações que se apresentam de seguida, no capítulo 7, as quais representam o novo sistema *SCADA*.

7. MÓDULOS DO SISTEMA

Neste capítulo descrevem-se as estruturas das aplicações tendo em conta as arquiteturas definidas para cada uma, detalhes gerais de implementação e o resultado final de cada uma das aplicações desenvolvidas.

7.1. MÓDULO APLICAÇÃO SERVIDOR

O módulo da aplicação servidor tem a responsabilidade de fazer a ponte entre o *hardware* (rede de sensores) e o *software*, possibilitando a aquisição dos dados adquiridos das redes de sensores e posterior tratamento desses dados.

7.1.1. Estrutura da Aplicação

A arquitetura definida para a implementação do módulo da aplicação servidor foi a arquitetura em “N camadas”. Na figura 18, descreve-se a estrutura definida para este módulo.

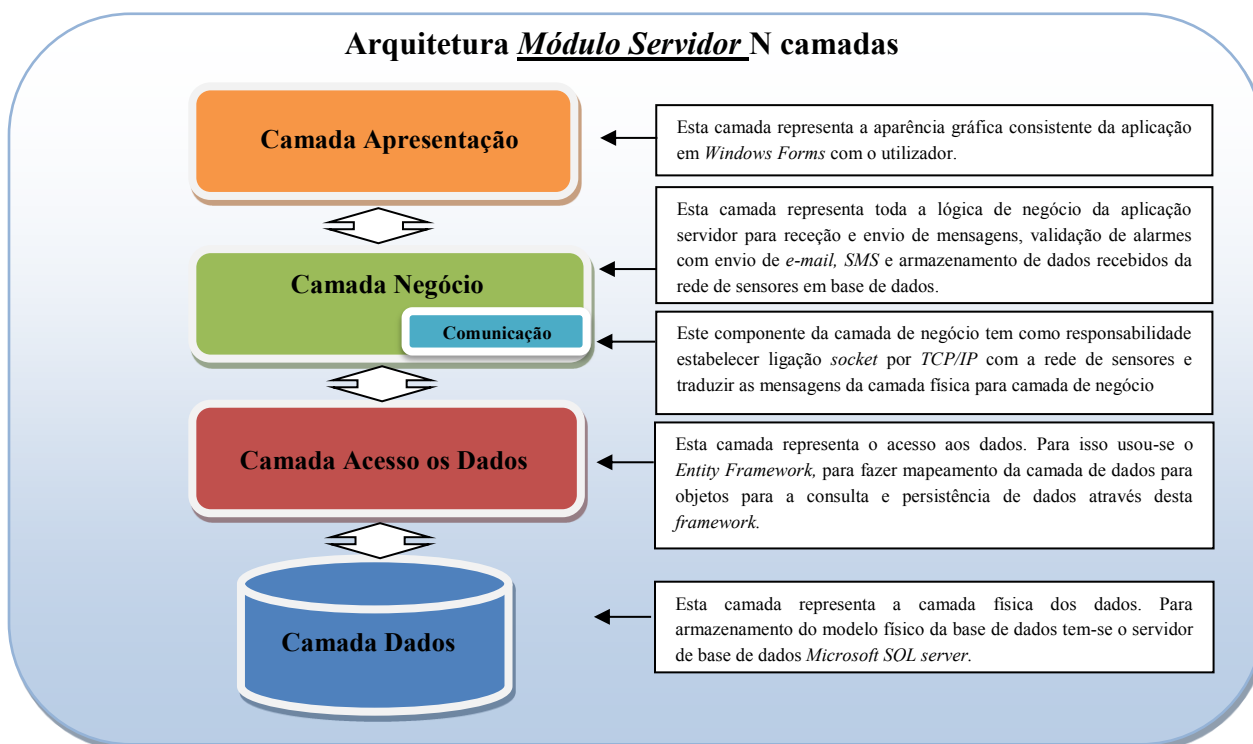


Figura 18 - Diagrama representativo da estrutura do módulo da aplicação servidor na arquitetura “N Camadas”

- **Camada de Apresentação** – Para desenvolver a camada gráfica da aplicação servidor criaram-se as diferentes janelas na aplicação com recurso a componentes *Windows Forms*;
- **Camada de negócio** – Na figura 19 é apresentado o diagrama de atividades onde é exposta a lógica de funcionamento da aplicação servidor. Para desenvolver esta camada usou-se a seguinte lógica de negócio: após ser configurado o *IP*, porto, e ativada ou desativada a opção verificar alarmes, o servidor é ligado, ficando a aguardar ligação das redes de sensores, estabelecendo-se a ligação por *socket* para transmissão e receção de mensagens. Em cada mensagem recebida são usados eventos e *threads* (TechTerms, 2014) para tratar as mesmas de forma assíncrona, passando pelas seguintes verificações:
 - Cada mensagem recebida tem o *id* da rede de onde provém. Por sua vez, o *id* dessa rede pertence sempre a uma empresa. O servidor, quando é ligado, tem sempre em memória a correspondência *id* da rede que pertence a uma empresa e atualiza-a periodicamente.
 - Se a mensagem recebida não pertencer a uma empresa, ou se o *id* do sensor não tiver sido configurado, é gerado um erro, que é guardado na base de dados na tabela de erros para futura consulta na aplicação *SCADA Web* pelo administrador.
 - Se a mensagem recebida pertencer a uma empresa é guardada na tabela *log*. Posteriormente é verificado se existe alarme nessa mensagem; caso exista, é guardada informação de alarme na tabela *log alarmes*.
 - Se o servidor tiver opção a enviar alarmes ativa, é enviado *e-mail* e *SMS* sempre que ocorra alarme.
 - Sempre que a aplicação *SCADA Web* enviar uma mensagem ao servidor, este encaminha essa mensagem para a rede quando esta estiver ligada.
- **Camada de Acesso aos Dados** – Para desenvolver esta camada de acesso aos dados usou-se o *Entity Framework*, para consulta, mapeamento e persistência de dados para objetos através desta *framework*;
- **Camada de Dados** – Esta camada representa a camada física onde a base de dados está configurada. Neste caso está configurada no servidor de base de dados *Microsoft SQL server*. Tendo o servidor de base de dados devidamente configurado para funcionar em rede, nesta aplicação existe um ficheiro *XML* que contém todas as configurações como *string* conexão para acesso ao servidor de base de dados.

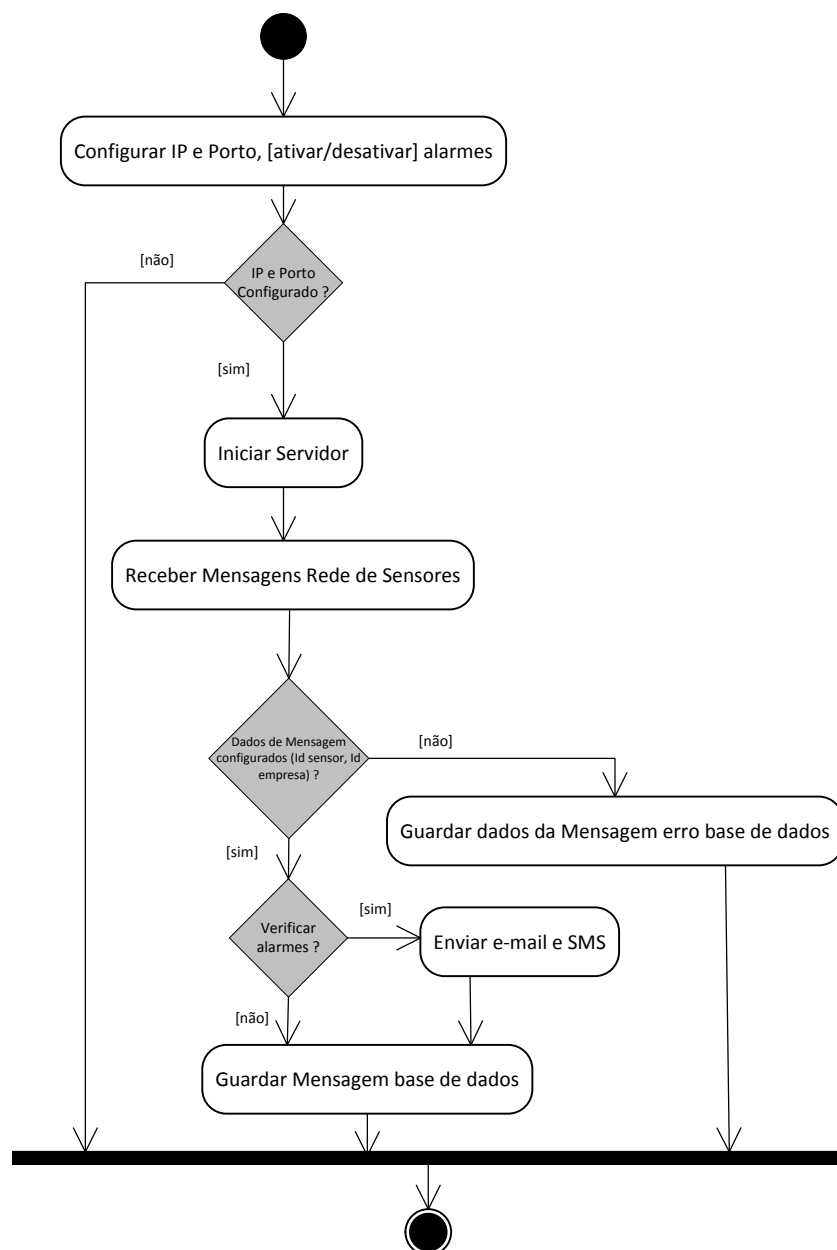


Figura 19 – Diagrama de atividades da aplicação servidor

7.1.2. Detalhes de Implementação

Nesta secção apresenta-se de uma forma genérica a implementação da aplicação servidor. Na figura 20, apresenta-se a solução da aplicação servidor onde se pode ver a estrutura física da aplicação. Esta aplicação foi desenvolvida em tecnologia *Windows Forms*, assente na *Framework .NET 4.0*, com uso específico da classe *System.Threading* para funcionamento assíncrono da aplicação e a classe *System.Net.Sockets*, para utilização da comunicação por *socket TCP/IP* da aplicação servidor com a rede de sensores.

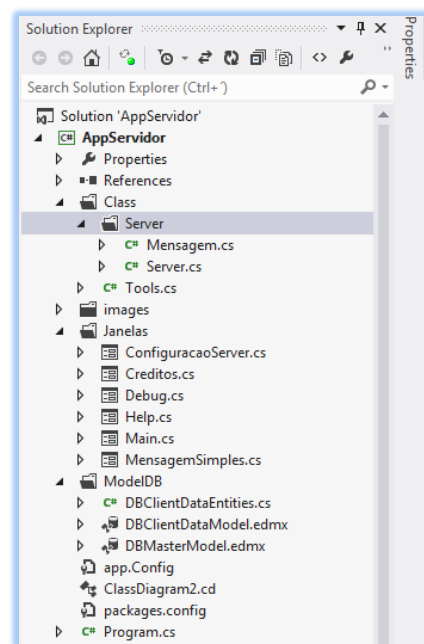


Figura 20 - Estrutura física da aplicação servidor no Visual Studio 2012.

A classe *Server.cs* vista na figura 20 é o objeto que contém todos os métodos e eventos necessários para criar servidor de *socket* por *TCP/IP*. Esta classe contém os métodos para criar servidor com *IP* e porto, desligar servidor, atender clientes, receber mensagens das redes de sensores, terminar ligação com clientes, entre outros. A classe *Server.cs* implementa a classe *Mensagem.cs*, objeto que contém todos os atributos necessários, para a estrutura das mensagens de dados usada pela *eneidaws*. Esta classe *Server.cs* contém métodos e eventos necessários para implementar a lógica de negócio do servidor de *sockets* na classe principal *Main.cs*.

Exemplo de alguns métodos principais existentes na classe *Server.cs*

```
int IniciaServidor(string IP, int porto){...}  
void TerminaServidor(){...}  
void _ProcessaMensagens(SocketClient client){...}
```

Exemplo de alguns eventos principais existentes na classe *Server.cs*

```
delegate void ChegouNovaMensagem(object sender, Mensagem mensagem);  
delegate void PedidoNovaLigacao(object sender, string IP);  
delegate void TerminaLigacao(object sender, string IP);
```

A classe *Tools.cs*, vista na figura 20, é o objeto que contém métodos para: enviar *e-mail* e *SMS*; consultar alarmes; verificar pelo *id* da rede a que empresa corresponde o alarme; obter contatos *e-mail* e telefone da empresa que tem a rede em alarme para envio de *e-mail* e *SMS*; verificar a mensagem recebida pelo *id* da rede de sensores e perceber a que empresa pertence para guardar informação da mensagem na base de dados do cliente, entre outros. Esta classe *Tools.cs* contém métodos necessários para implementar a lógica de negócio na classe principal *Main.cs*.

Exemplos de alguns métodos principais existentes na classe *Tools.cs*

```
void getListaEmpresaIDRedes(){...}

string verificaIDRedeEmpresa(int idRede){...}

void verificaAlarmeCritico(DBClientDataEntities db, int IRRede,
int IDSensor, int codFuncao, int valorLido, string Identificador){...}

void consultaAlarmesBD(DBClientDataEntities db){...}

void enviarEmailAlarme(DBClientDataEntities db,
List<Log_MensagensAlarmeServidor> alarmes){...}

string HTTPSendSMS(string sHTTPInputData) {...}
```

A pasta *Janelas*, vista na figura 20, contém todas as janelas definidas para a aplicação: a janela principal da aplicação e as janelas filhas. A pasta *ModelDB*, vista na figura 20, contém o modelo *object relational mapping (ORM)*, obtido através da base de dados física com *Entity Framework 5.0*. O ficheiro *app.config*, em formato *XML*, contém as configurações da aplicação como versão *framework .NET* a ser usada na aplicação e a *string* de conexão para o servidor de base de dados.

A classe *Program.cs*, vista na figura 20, é classe principal, porque aqui é onde se inicia a aplicação. Quando a aplicação é executada, a classe *Program.cs* chama a *form* principal da aplicação, neste caso a *form Main.cs*. A classe *Main.cs* contém toda a lógica da aplicação, onde são executados os processos: receção de mensagens da rede de sensores e armazenamento dos dados na base de dados; envio de mensagem à rede de sensores; verificação da existência de alarmes e respetivo envio de alarmes por *e-mail* e *SMS*.

A figura 21 representa o diagrama de classes do modelo que é usado pela aplicação servidor para fazer consulta e persistência de dados. Os objetos principais usados por esta aplicação são: a classe *Log_MensagemServidor*; a classe *Log_MensagemAlarmeServidor*; a classe *Log_MensagemErroServidor*, para guardar os dados das mensagens referentes à classe *Entrada*, que está relacionada com a classe *Sensor*, a qual, por sua vez, está relacionada com a classe *Rede*. Outra classe que é importante no uso desta aplicação é a classe *Empresa_IDRede*. Esta classe permite ao servidor saber que uma determinada mensagem recebida com o *id* de uma rede pertence a uma empresa e, em *runtime*, o servidor estabelece uma ligação à base de dados do cliente dessa empresa para armazenar os dados recebidos.

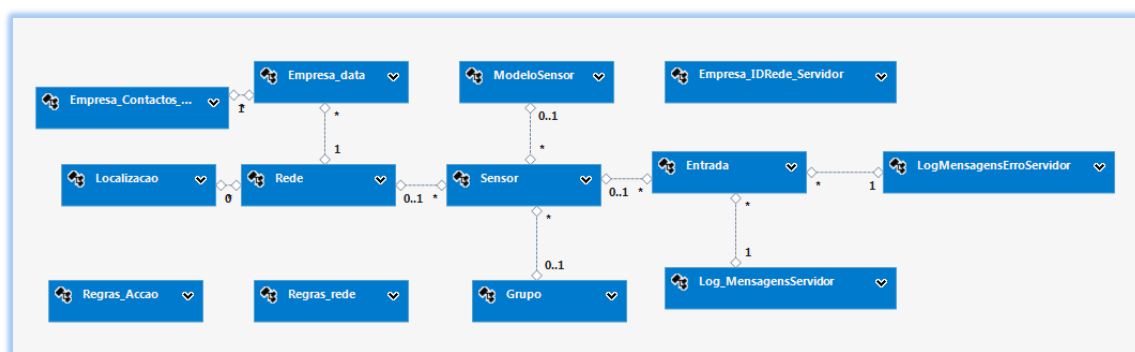


Figura 21 - Diagrama de Classes²⁷ do modelo mapeado pelo Entity Framework

Em suma, a camada de apresentação é definida pela pasta *Janelas*, onde estão definidos os diferentes objetos gráficos da aplicação em *Windows Forms*. A camada de negócio é definida para a classe *Main.cs*, que implementa o objeto *Server.cs*, responsável por criar o servidor de *sockets* por *TCP/IP*, para comunicação com a rede de sensores. A classe *Main.cs* implementa também a classe *Tools.cs*, que contém repositório de métodos genéricos, como enviar *e-mail* e *SMS*, verificar alarmes, entre outros. A classe *Main.cs* contém todo o funcionamento da aplicação servidor com a receção assíncrono por cada cliente atendido. A camada de acesso aos dados é feita através da pasta *ModelDB*, onde o *Entity Framework* mapeia as entidades da base de dados para objetos, como podemos ver no digrama de classes da figura 21. A camada de dados é definida pela base de dados física.

7.1.3. Aplicação Servidor

A aplicação servidor tem como objetivo apresentar um *design* simples. Esta aplicação é um executável que se instala localmente num servidor *Windows* para fazer tratamento e processamento de mensagens, sendo esta configurada exclusivamente pelo administrador da empresa *eneidaws*. Na figura 22 apresenta-se o aspeto gráfico da aplicação servidor. Quando esta aplicação é minimizada, funciona em *background*, ficando exclusivamente visível na barra de tarefas como se pode verificar na figura 23, bastando clicar no ícone para maximizar. No Anexo D, pode ser consultado o manual de utilizador desta aplicação.

²⁷ Os objetos dos diagramas de classes apresentados neste capítulo não têm os atributos visíveis, para que as imagens dos diagramas de classes não ocupem muito espaço. No entanto, no Anexo C, podem ser vistos os atributos destes objetos.

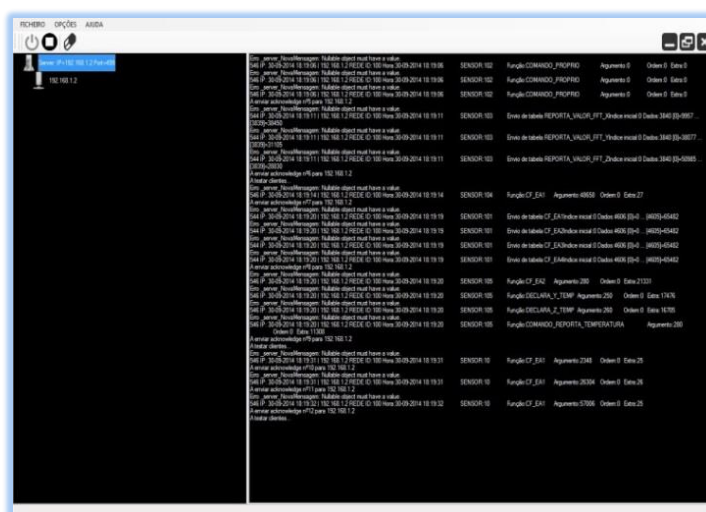


Figura 22 - Aplicação Servidor



Figura 23 - Ícone da aplicação servidor na barra de tarefas

7.2. MÓDULO APLICAÇÃO SERVIÇO DE ALARMES

O módulo da aplicação serviço de alarmes tem a responsabilidade de verificar a existência de alarmes. Em caso de existência de um alarme, esta aplicação deverá notificar o utilizador por *e-mail* e *SMS*.

7.2.1. Estrutura da Aplicação

A arquitetura definida para desenvolvimento do módulo da aplicação serviço de alarmes foi a arquitetura em “N camadas”. Na figura 24, descreve-se a estrutura definida para este módulo.

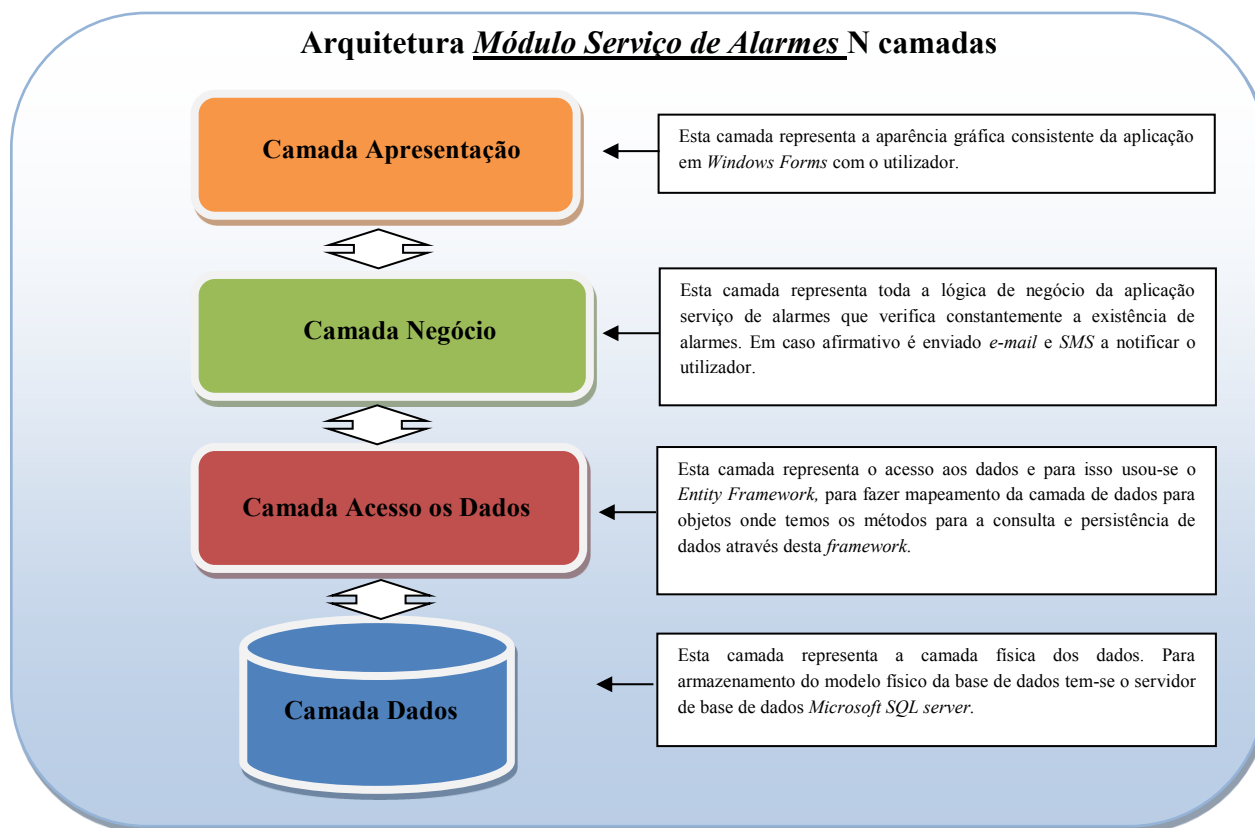


Figura 24 - Diagrama representativo da estrutura do módulo da aplicação serviço de alarmes na arquitetura "N Camadas"

- **Camada de Apresentação** – Para desenvolver a camada gráfica da aplicação serviço de alarmes criaram-se as diferentes janelas da aplicação com recursos a componentes *Windows Forms*;
- **Camada de negócio** – Na figura 25 é apresentado o diagrama de atividades onde é exposta a lógica de funcionamento da aplicação serviço de alarmes. Para desenvolver esta camada usou-se a seguinte lógica de negócio.
 - Após ser configurado o relógio da aplicação serviço de alarmes, este serviço é colocado em execução.
 - De acordo com o relógio configurado, é feita uma análise periódica à tabela que contém todos os alarmes detetados pelo servidor e guardados na tabela *log alarmes*.
 - Quando é feita a verificação da tabela de alarmes é verificado se o alarme já foi enviado ou não.
 - Caso o alarme ainda não tenha sido enviado é verificada a empresa a que corresponde a mensagem. Por sua vez, são verificados os utilizadores da empresa que estão definidos para receber notificações.

- Após o envio da notificação de alarme por *e-mail* e *SMS*, é marcado na base de dados que essa mensagem de alarme já foi enviada.
- O processo de análise e envio de *e-mails* e *SMS* é feito de forma assíncrona com o recurso a *threads*.
- **Camada de Acesso aos Dados** – Para desenvolver esta camada de acesso aos dados usou-se o *Entity Framework* para consulta, mapeamento e persistência de dados para objetos através desta;
- **Camada de Dados** – Esta camada representa a camada física onde a base de dados está configurada; neste caso, está configurada no servidor de base de dados Microsoft *SQL Server*. Tendo o servidor de base de dados devidamente configurado para funcionar em rede, nesta aplicação existe um ficheiro *XML* que contém todas as configurações como *string* conexão para acesso ao servidor de base de dados.

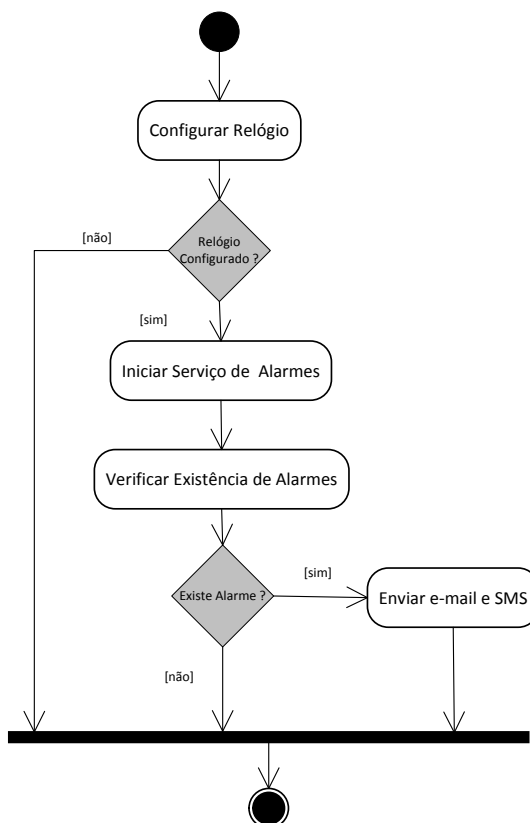


Figura 25 – Diagrama de atividades da aplicação serviço de alarmes

7.2.2. Detalhes de Implementação

Nesta secção apresenta-se de uma forma genérica a implementação da aplicação serviço de alarmes. Na figura 26, apresenta-se a solução da aplicação serviço de alarmes onde se pode ver a estrutura física da aplicação. Esta aplicação foi desenvolvida em tecnologia *Windows Forms*,

assente na *Framework .NET 4.0*, com uso específico da classe *System.Threading* para funcionamento assíncrono da aplicação.

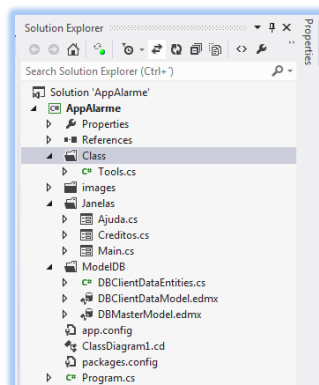


Figura 26 - Estrutura física da aplicação Serviço de Alarmes no Visual Studio 2012

A classe *Tools.cs*, vista na figura 26, é um objeto que contém métodos para: enviar *e-mail* e *SMS*, criar relógio, consultar alarmes, verificar pelo *id* da rede a que empresa corresponde o alarme, obter contactos *e-mail* e telefone da empresa que tem a rede em alarme para envio de *e-mail* e *SMS*, entre outros. Esta classe contém todos os métodos necessários para implementar a lógica de negócio na classe principal *Main.cs*.

Exemplos de alguns métodos principais existentes na classe *Tools.cs*

```
int IniciaServico(string tipoTempo, int tempo){...}

void TerminaServidor(){...}

void enviarEmailAlarme(DBClientDataEntities db,
List<Log_MensagensAlarmeServidor> alarmes){...}

string HTTPSendSMS(string sHTTPInputData) {...}
```

A camada de apresentação é representada pela pasta *Janelas*, vista na figura 26, contém todas as janelas definidas para a aplicação: a janela principal da aplicação e as janelas filhas. A pasta *ModelDB*, vista na figura 26, contém o modelo *ORM*, obtido através da base de dados física com *Entity Framework 5.0*. O ficheiro *app.config*, em formato *XML*, contém as configurações da aplicação como versão *framework .NET* a usar na aplicação e a *string* de conexão para o servidor de base de dados.

A classe *Program.cs* vista na figura 26 é a classe principal porque é aqui onde se inicia a aplicação. Quando a aplicação é executada, a classe *Program.cs* chama a *form* principal da aplicação, neste caso a *form Main.cs*. A camada de negócio é representada pela classe *Main.cs*, que implementa a classe *Tools.cs*, contendo toda a lógica da aplicação, onde é executado o processo verificação de alarmes, que funciona de forma assíncrona, e consulta periódica aos objetos do modelo visto na figura 27. A camada de dados é definida pela base de dados física.

Na figura 27 apresentam-se as classes do modelo que são usadas pela aplicação serviço de alarmes para fazer consulta e persistência de dados. Os objetos usados principalmente por esta aplicação são a classe *Empresa_IDRede*, contém a associação do *id* da rede que pertence a uma empresa. A classe *Empresa_Contactos_Alarme*, corresponde ao contacto telefónico e *e-mail* dos utilizadores de uma empresa para o envio de *e-mail* e *SMS*. A classe *Log_MensagensAlarmeServidor* disponibiliza toda a informação das redes de sensores que têm alarmes, sendo esta classe consultada periodicamente.

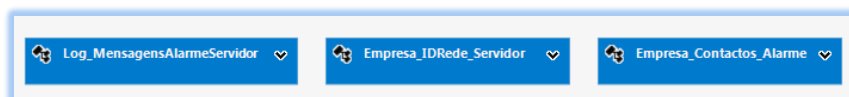


Figura 27 - Diagrama de Classes do modelo mapeado pelo Entity Framework

7.2.3. Aplicação Serviço de Alarmes

A aplicação serviço de alarmes tem como objetivo apresentar um *design* simples. Esta aplicação é um executável que se instala localmente num servidor para fazer validação de alarmes existentes nas redes de sensores, sendo configurada exclusivamente pelo administrador da *eneidaws*. Na figura 28, apresenta-se o aspeto gráfico da aplicação serviço de alarmes. Quando esta aplicação é minimizada, funciona em *background*, ficando exclusivamente visível na barra de tarefas, como se pode ver na figura 29; para maximizar, basta clicar no ícone. No Anexo D, pode ser consultado o manual de utilizador desta aplicação.

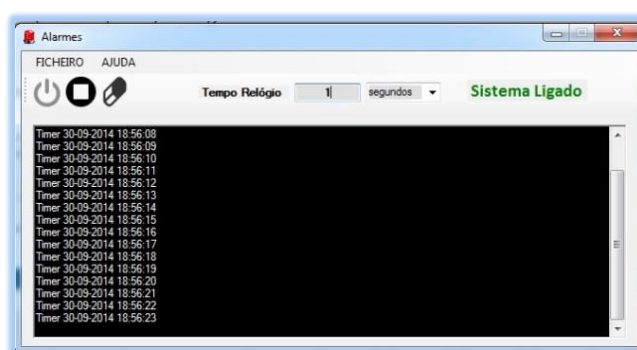


Figura 28 - Design da Aplicação Serviço de Alarmes



Figura 29 - Ícone da aplicação Serviço de alarmes na barra de tarefas

7.3. MÓDULO APLICAÇÃO *WEB SERVICE*

O módulo da aplicação *Web Service* tem como responsabilidade disponibilizar informação das redes de sensores através de pedidos *HTTP*, por *GET* e *POST*.

7.3.1. Estrutura da Aplicação

A arquitetura definida para desenvolvimento do módulo da aplicação *Web Service* foi a arquitetura *Windows ASP.NET Web API*. Na figura 30, descreve-se a estrutura definida para este módulo.

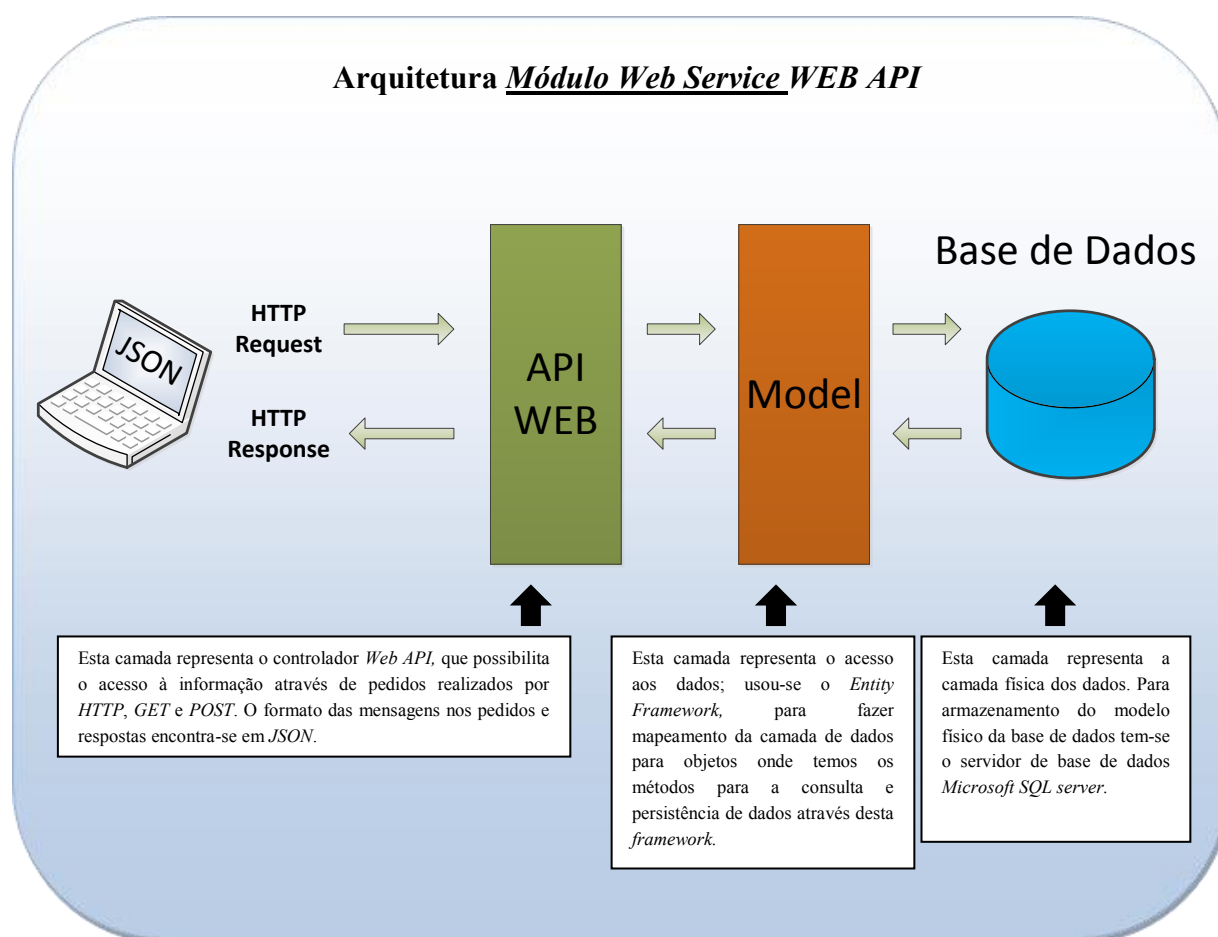


Figura 30 - Diagrama representativo da estrutura do Módulo Web Service

- **Camada *WEB API*** – A camada *Web API*, representa o controlador, que possibilita o acesso à informação através de pedidos realizados por *HTTP*, *GET* e *POST*. Neste controlador estão implementados todos os métodos públicos que permitem, a

disponibilização de informação das redes de sensores ao utilizador. A informação só é disponibilizada usando uma chave válida fornecida pelo administrador da *eneidaws*, caso contrário o *Web Service* não disponibiliza informação. O formato das mensagens trocadas nos pedidos e respostas encontra-se em *JSON*.

- **Camada Modelo** – Para desenvolver esta camada de acesso aos dados usou-se o *Entity Framework* para consulta, mapeamento e persistência de dados para objetos através desta;
- **Camada de Dados** – Esta camada representa a camada física onde a base de dados está configurada; neste caso está configurada no servidor de base de dados Microsoft *SQL Server*.

7.3.2. Detalhes de Implementação

Nesta secção apresenta-se de uma forma genérica a implementação da aplicação *Web Service*. Na figura 31, apresenta-se a solução da aplicação *Web Service* onde se pode ver a estrutura física da aplicação. Esta aplicação foi desenvolvida em tecnologia *ASP.NET Web API*, assente na *Framework .NET 4.5*.

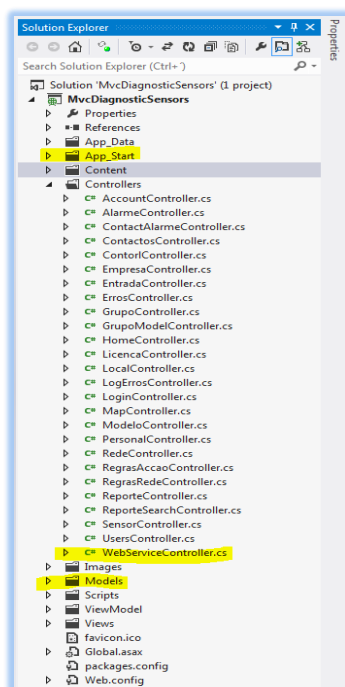


Figura 31 - Estrutura física da aplicação *Web Service* no Visual Studio 2012

Como se pode ver na figura 31, o desenvolvimento da aplicação *Web Service* foi integrado na estrutura da solução *MVC* da aplicação *SCADA Web*. Esta situação verifica-se porque a estrutura de desenvolvimento e funcionamento da tecnologia *ASP.NET Web API* é semelhante à tecnologia *ASP.NET MVC*. Em ambas as tecnologias existe o modelo e controlador. No *ASP.NET MVC* o retorno da informação é feito para uma vista/página *Web* e o retorno do controlador no *ASP.NET Web API* é feito no formato *JSON*; nesta aplicação poderia ser também usado o formato *XML*.

Para desenvolver a aplicação baseada em *Web Service* foi necessário configurar a classe *WebApiConfig.cs* existente na pasta *App_Start*, vista na figura 31, e implementar o controlador para este serviço *Online*.

A classe *WebApiConfig.cs* tem como responsabilidade a configuração das rotas nos pedidos feitos por *HTTP*, como se pode ver num exemplo do excerto de código desta classe. Caso essa rota não exista nesta classe, é enviada uma resposta de erro por parte do *Web Service*.

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.Routes.MapHttpRoute(
            name: "GetAllLocalizacoes",
            routeTemplate: "api/{controller}/GetNumLastMensagens/{key}/{IdSensor}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}
```

Na secção 7.3.3, exemplifica-se como é realizado um pedido ao *Web Service*, tendo em conta as rotas configuradas no *WebApiConfig.cs*.

A classe *WebServiceController.cs* representa o controlador criado para a aplicação *Web Service*. Neste controlador estão implementados todos os métodos *GET* e *POST*, para disponibilização da informação relativa à rede de sensores. Este controlador obtém toda a informação a partir do digrama de classes do modelo através da consulta dos dados.

Exemplo de alguns pedidos existentes no controlador *WebServiceController.cs*

```
IQueryable<object> GetAllLocalizacoes(int key, int id){...}
IQueryable<object> GetAllRedes(int key, int id)
List<Log_MensagensServidor> GetAllMensagens(int key, int sensor)
List<Log_MensagensAlarmeServidor> GetAllMensagensAlarme(int key, int sensor)
List<LogMensagensErroServidor> GetAllMensagensErro(int key, int id)
```

Na figura 32 apresentam-se as classes do modelo que são usadas pela aplicação *Web Service* para fazer consultas e persistência de dados. Os objetos usados por esta aplicação são as classes *Localização*, *Rede*, *Sensor*, *Grupo*, *ModeloSensor* e *Entrada*.

Através do relacionamento entre estes objetos é obtida informação das redes de sensores através das classes *Log_MensagemServidor*, *Log_MensagemErroServidor* e *Log_MensagemAlarmeServidor*, que contêm os dados recolhidos na base de dados.

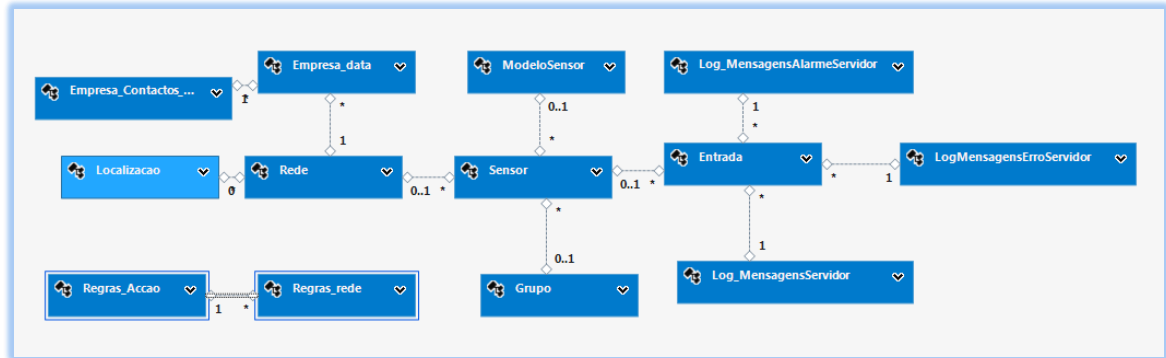


Figura 32 - Diagrama de Classes do modelo mapeado pelo Entity Framework

7.3.3. Pedido *HTTP* Aplicação Web Service

Nesta secção exemplifica-se como pode ser realizado um pedido *GET* à aplicação *Web Service*, com retorno da informação no formato *JSON*. No Anexo D, pode ser consultado o manual de utilizador desta aplicação, com todos os pedidos que podem ser realizados ao *Web Service*.

Exemplo pedido *GET*

Aqui exemplifica-se como pode ser construído um pedido *HTTP GET* ao *Web Service* que retorne todos os últimos dados adquiridos para um sensor com um determinado id.

Pedido público [*http://domino.net/api/webservice/*] [*Nome Método*] [*Chave*] [*Id Sensor*]

<http://domino.net/api/webservice/GetNumLastMensagens/CHAVE/IDSENSOR/>

```
- {
  id_evento: 56055,
  id_empresa: 2,
  id_entrada: 1,
  IDRede: 100,
  IDSensor: 10,
  codigofuncao: 11,
  valor: null,
  valor_argumento: 399,
  valor_extra: null,
  valor_ordem: null,
  data: "2014-09-01T01:51:37",
  sequencia: null
},
```

Figura 33 - Resultado Pedido GET em JSON

7.4. MÓDULO APLICAÇÃO SCADA WEB

O módulo da aplicação *SCADA Web* representa a aplicação *SCADA* para supervisão e controlo de todas as redes de sensores instaladas nas diferentes unidades industriais.

7.4.1. Estrutura da Aplicação

A arquitetura definida para desenvolvimento do módulo da aplicação *SCADA Web* foi a arquitetura *MVC* desenvolvida em tecnologia *ASP.NET MVC*. Na figura 34, descreve-se a estrutura definida para este módulo e os principais componentes desta aplicação como se pode ver na figura 35, que correspondem ao módulo da aplicação *SCADA Web*.

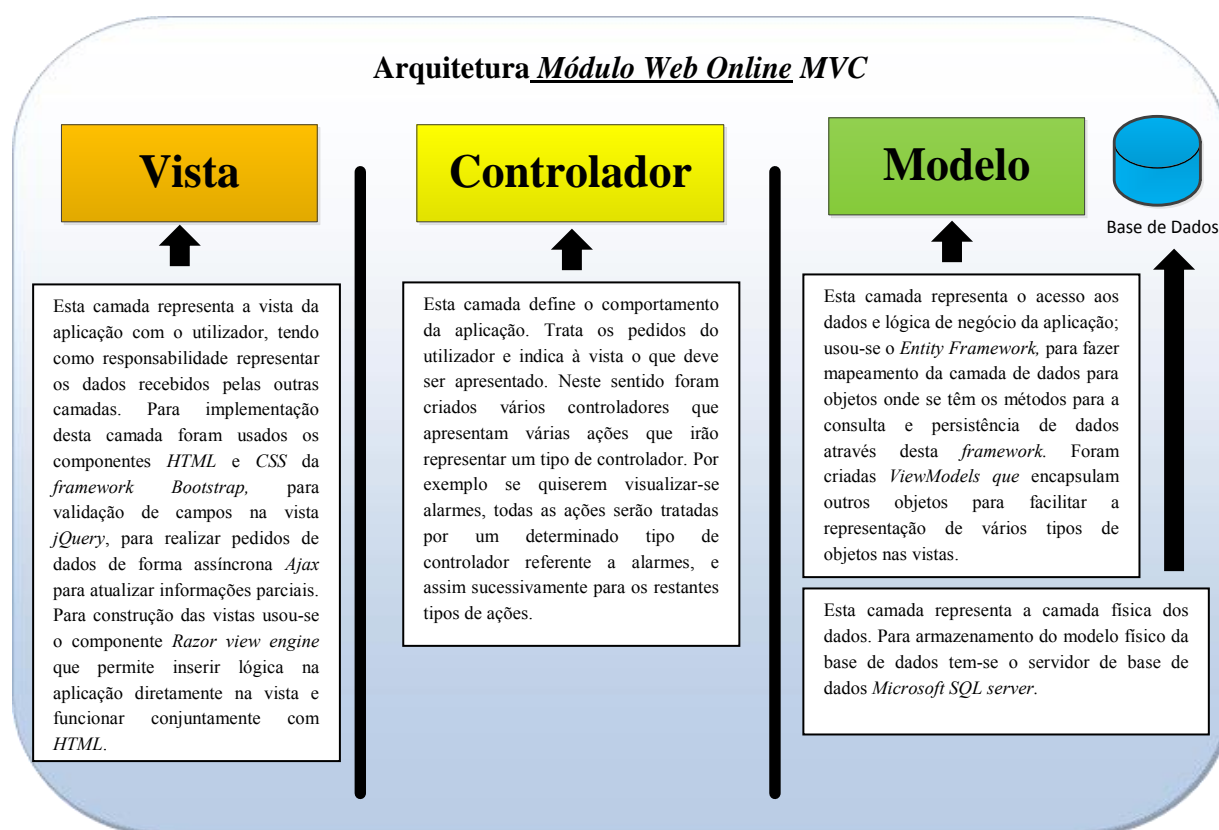


Figura 34 - Diagrama representativo da estrutura do módulo da aplicação *SCADA Web*

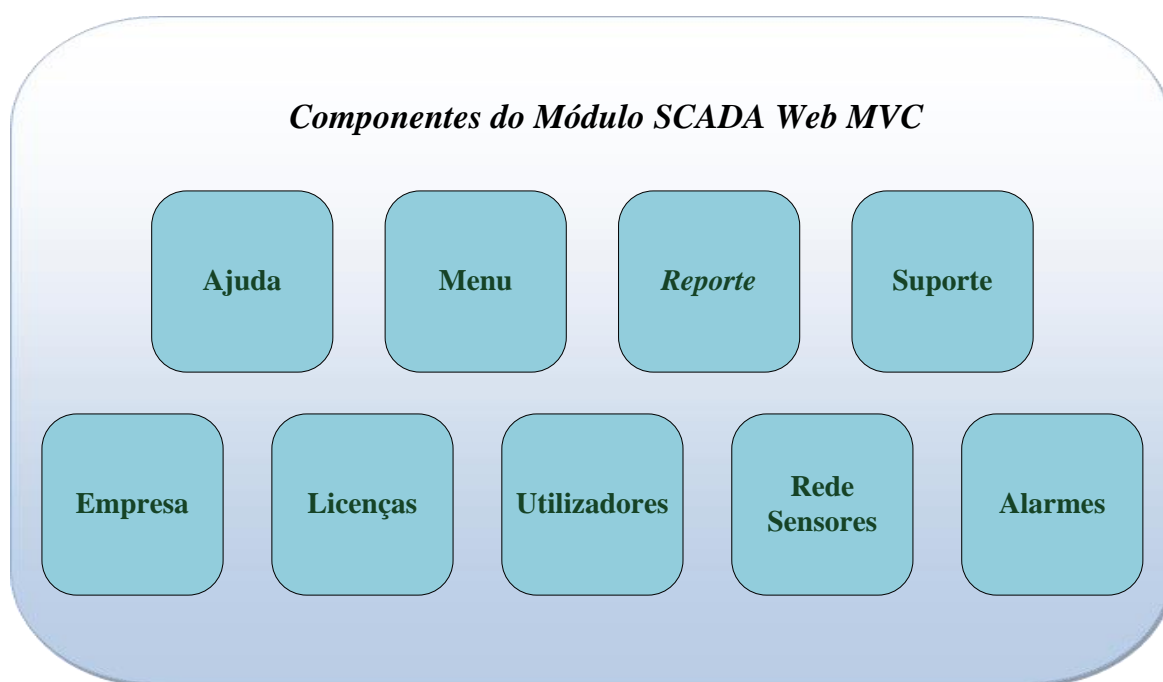


Figura 35 - Diagrama representativo da estrutura de componentes principais do módulo da aplicação SCADA Web

Para elaboração do módulo aplicação *SCADA Web*, subdividiu-se este em diferentes componentes principais que representam as funcionalidades principais da aplicação *SCADA Web*, vistos na figura 35, com o objetivo de tornar a estrutura da aplicação modelar e flexível para adição, atualização ou remoção de novas funcionalidades.

A implementação de cada um destes componentes principais corresponde ao desenvolvimento de um controlador. Cada um destes controladores pode relacionar-se com outros controladores para partilha de informação. De seguida apresenta-se uma breve explicação da lógica implementada para cada um dos componentes.

- **Componente Ajuda** – Este controlador foi criado para existir uma vista com tópicos de ajuda para os utilizadores poderem esclarecer as suas dúvidas referentes à aplicação. Este controlador está disponível no menu “Ajuda”, representado pelo ícone ponto de interrogação.
- **Componente Menu** – Este controlador representa os diferentes menus que são apresentados aos utilizadores. Com a existência de controlo de acessos, em que cada utilizador tem diferentes níveis de privilégios e regras de acesso, este componente permite de forma automática apresentar os diferentes menus a que o utilizador tem acesso de forma dinâmica. Por defeito, cada controlador tem definido, com base nas regras de acesso, que utilizadores podem ter acesso às ações do respetivo controlador.

- **Componente Suporte** – Este controlador permite aos utilizadores da plataforma usarem este componente para visualizar, criar e apagar mensagens. O menu “Suporte” está disponível para todos os utilizadores.
- **Componente Empresa** – Este controlador permite ao utilizador administrador da *eneidaws*, gerir, criar, editar e remover empresas da aplicação *Web*. Este controlador está disponível no menu “Administração”.
- **Componente Licenças** – Este controlador permite ao utilizador administrador da *eneidaws* gerir, criar, editar e remover licenças da aplicação *Web*, associando cada tipo de licença (normal ou completa) a uma empresa. Este controlador está disponível no menu “Administração”.
- **Componente Utilizadores** – Este controlador permite ao utilizador administrador da *eneidaws* gerir, criar, editar e remover utilizadores, bem como escolher o seu nível de acesso na aplicação *Web*. Este controlador está disponível no menu “Administração”; no entanto, está igualmente disponível para o cliente da *eneidaws* que é administrador, criando os utilizadores associados à sua empresa, e escolhendo quais dos seus utilizadores recebem alarmes existentes numa rede de sensores, estando disponível no menu “Configuração”.
- **Componente Reporte** – Este controlador está disponível para todos os utilizadores das empresas que tenham a licença completa, no menu “Reporte”. Permite analisar todas as redes de sensores através de histórico de dados de alarmes ou através de gráficos, tabelas. Há a possibilidade de pesquisa de informação mais específica e a opção de exportar estas informações para *Excel*, *PDF*, *CSV*, etc.
- **Componente Alarmes** – Este controlador está disponível para visualização dos últimos alarmes ocorridos, para todos os utilizadores no menu “Alarmes”, existindo a particularidade de o utilizador com privilégios elevados marcar o alarme como resolvido adicionando uma observação. Este controlador permite ainda gerir, criar, editar e remover regras para alarmes, criadas para as redes sensores estando disponível para o cliente da *eneidaws*, no menu “Configuração”.
- **Componente Rede de Sensores** – Este controlador foi criado para criar, gerir, editar e desativar, importar e exportar rede de sensores. Cada rede de sensores relaciona-se com os controladores sensor, modelo sensor, entrada, grupo e localização, os quais têm igualmente ações criar, gerir, editar e remover. Em termos lógicos uma rede de sensores é representada pela relação destes controladores. Este controlador está disponível no menu “Configuração”, para o cliente da *eneidaws*.

7.4.2. Detalhes de Implementação

Nesta secção apresenta-se de uma forma genérica a implementação da aplicação *SCADA Web*. Na figura 36, apresenta-se a solução da aplicação *SCADA Web* podendo ver-se a estrutura física desta aplicação. Esta aplicação foi desenvolvida em tecnologia *ASP.NET MVC* assente na *Framework .NET 4.5*, entre outras tecnologias usadas, como abordado no capítulo 5.

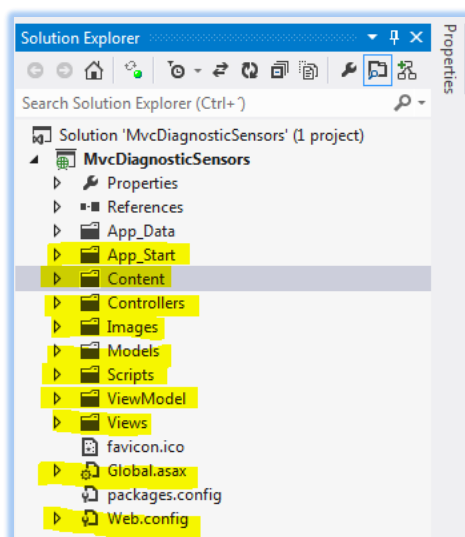


Figura 36 - Estrutura física da aplicação *SCADA Web* no Visual Studio 2012

Para desenvolvimento desta aplicação *SCADA Web*, cada funcionalidade e opção (vistas na figura 35) corresponde a um ou mais controladores, permitindo que todos estes funcionem de forma independente.

Na figura 36 encontra-se a estrutura física criada para cada funcionalidade e opção, que corresponde a um ou mais controladores, os quais contêm as suas próprias vistas e respetivo acesso aos modelos presentes nas figuras 37 e 38.

O acesso aos controladores é feito através da classe *RouteConfig* que se encontra na pasta *App_Start*, vista na figura 36, existente na *framework ASP.NET MVC*. Esta classe tem como responsabilidade tratar rotas para fazer requisições aos controladores referentes a uma determinada ação (por exemplo listar redes de sensores). Esta regra, vista no excerto de código da classe *RouteConfig*, deve-se ao facto de a estrutura da rota estar registada na classe e define-se o *url*: "{controller}/{action}/{id}". Caso o nome do *controller* e a *action* não estejam corretos no endereço *url*, a aplicação assume os valores por defeito “*Home*” para *controller* e “*Index*” para *action*, tratando-se dos conteúdos que aparecem na página inicial.

Exemplo de um *link* para listar informação sobre rede de sensor:

<http://diagnosticsoftware.dtdns.net/rede/1>

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        routes.MapRoute(
            name: "Default",
            url: "{controller}/{action}/{id}",
            defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }
        );
    }
}
```

A pasta *Controller*, vista na figura 36, contém todos os controladores desenvolvidos e cada um tem as suas próprias funcionalidades, cujos componentes são representados na figura 35. Nestes controladores são executadas várias ações que permitem ao controlador ter acesso aos modelos para consulta e persistência de dados. As ações genéricas que se podem encontrar nestes controladores são: criar, remover, consultar, atualizar e listar dados, entre outras ações mais específicas. Cada uma das ações tem que ter os pedidos *GET* para obter informações e pedidos *POST* para submeter dados; isto verifica-se para cada ação existente no controlador.

Exemplo de uma ação que pode ser encontrada num controlador

```
// GET: /Sensor/Create
public ActionResult Create(int id){...}

// POST: /Sensor/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create(RedeSensor model){...}
```

A pasta *View*, vista na figura 36, contém todas as vistas criadas, e cada uma dessas está associada a um único controlador, estando cada uma destas preparada para receber um tipo de objeto. Cada vista contém a representação gráfica de um controlador, permitindo desta forma a visualização dos dados de acordo com a ação executada. Nas vistas foram utilizadas diferentes tecnologias, como: uso de *Razor Engine View*, que permite criar as vistas dos controladores, uso de componentes da *framework Bootstrap* para o aspeto gráfico das vistas *HTML* e *CSS*, *jQuery* para validação de formulários e criar animações gráficas; foi usada também a tecnologia *Ajax* para fazer pedidos assíncronos de informação aos controladores, atualizando informação das vistas/páginas *Web*, sem fazer *refresh* às páginas *Web*.

Exemplo de um pedido *GET* em *Ajax* a solicitar ao controlador Rede a vista *index da Rede*, atualizando o conteúdo do retorno no *container master*

```
$("#rede_cr").click(function () {
    $.ajax({
        cache: false,
        type: "GET",
        url: "/Rede/index",
        success: function (result) {
            $('#master').html(result);
        },
        error: function (xhr, ajaxOptions, thrownError) {
        }
    });
});
```

Exemplo de um pequeno excerto de código fonte de uma vista, onde se pode ver que esta vista suporta uma lista de objetos do tipo “Control”. Aqui pode ver-se o código *HTML*, conjuntamente com a sintaxe *Razor View Engine*, (por exemplo `@Html.ActionLink("Create New", "Create")`), após renderizado pelo servidor será um *link HTML* que representará a função criar controlo. As vistas em *Razor* permitem também a programação de código fonte em *C#* como se pode ver o ciclo *foreach* para listar todos os objetos recebidos na vista.

```
@model IEnumerable<MvcDiagnosticSensors.Models.Control>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table>
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.id_control)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.id_accao)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Username)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Role)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.AccaoDesc)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Estado)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.data)
        </th>
        <th></th>
    </tr>
    @foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.id_control)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.id_accao)
            </td>
            (...)
```

A pasta *ViewModels*, vista na figura 36, contém as classes criadas que encapsulam outros objetos, permitindo enviar diferentes tipos de objetos para as vistas. Como as vistas associadas a um controlador só permitem a receção de um tipo de objeto, é necessário criar objetos compostos. Por exemplo o *ViewModel* classe *RedeVista* criado contém três tipos de objetos obtidos a partir do modelo, classe *Rede*, *Sensor*, *Entrada*. Ao instanciar um objeto da classe *RedeVista* são instanciados todos os outros tipos de objetos e isso permite o envio de um único tipo de objeto que a vista suporta com a possibilidade de ter os outros tipos de objetos com outras informações.

A pasta *Models*, vista na figura 36, contém o diagrama de classes dos modelos com os objetos mapeados pelo *Entity Framework* a partir da base de dados geral, vista na figura 37 e a partir da base de dados rede de sensores dos clientes, vista na figura 38.

A pasta *Content*, *Images* e *Scripts*, vista na figura 36, contém a informação referente aos conteúdos do *Web site*. Por exemplo, *CSS*, imagens e scripts *jQuery*, entre outros.

O ficheiro *Global.asax*, visto na figura 36, é o ficheiro que invoca as classes existentes na pasta *App_Start*, cada vez que um cliente abre o *Web site* no *browser*.

O ficheiro *Web.config*, visto na figura 36, é um ficheiro *XML*, que contém todas as configurações associadas à aplicação *Web*. Por exemplo, neste ficheiro encontram-se as *strings* de conexão ao servidor de base de dados, a versão das *frameworks .Net* usadas, entre outros. Como o nome indica, é o ficheiro de configuração principal da aplicação.

Na figura 37 apresenta-se o diagrama de classes do modelo que é usado pela aplicação *Web* para fazer consultas e persistência de dados. Todos os objetos do modelo da figura 37 são usados, por exemplo, para validação das contas de *login* e regras de acesso, obter informação da empresa, obter informação dos utilizadores, entre outros. A classe *Empresa_IDRede* permite que o utilizador, ao fazer *login*, seja identificado e, mediante a sua identificação, este é redirecionado para os seus dados da empresa e da rede de sensores, onde é estabelecida a conexão à sua base de dados, sem que este se aperceba.

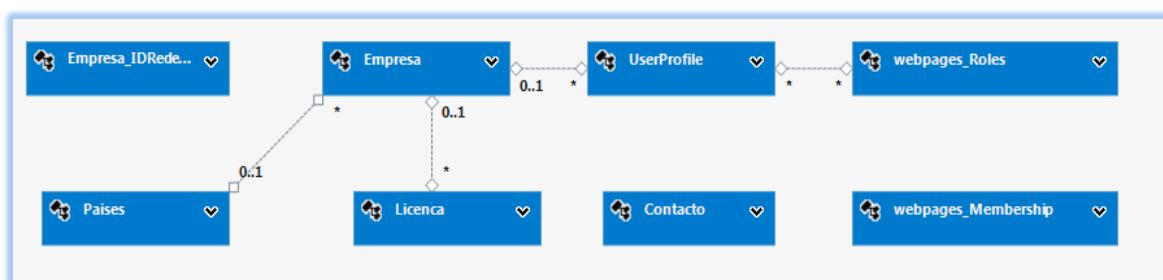


Figura 37 - Diagrama de Classes do modelo mapeado pelo Entity Framework

Na figura 38 apresenta-se o diagrama de classes do modelo que são usadas pela aplicação *Web* para fazer consultas e persistência de dados. Todos os objetos são usados por esta aplicação, sendo estas classes *Localização*, *Rede*, *Sensor*, *Grupo*, *ModeloSensor* e *Entrada*, etc., que contém toda a informação associada à rede de sensores dos vários clientes e respetivas configurações.

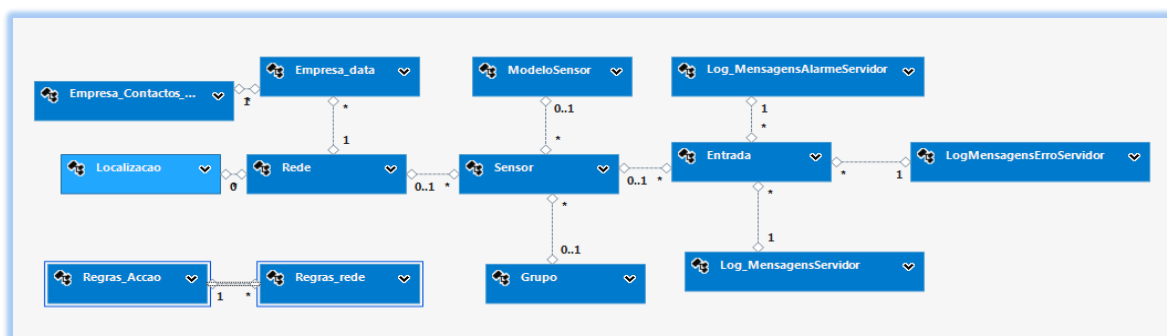


Figura 38 - Diagrama de Classes do modelo mapeado pelo Entity Framework

7.4.3. Aplicação SCADA Web

A aplicação *SCADA Web* tem como objetivo apresentar um *design* simples e *user friendly*, capaz de funcionar em diferentes dispositivos. Esta é uma aplicação *SCADA Web* que tem de ser alojada num servidor *Web IIS*, que pertence à plataforma *.NET*, existente nos sistemas operativos *Windows*.

Esta aplicação *SCADA* tem como responsabilidade a monitorização e diagnóstico das redes de sensores instaladas. Na figura 39, apresenta-se o aspeto gráfico da aplicação *SCADA Web*. No Anexo D, pode ser consultado o manual de utilizador desta aplicação.

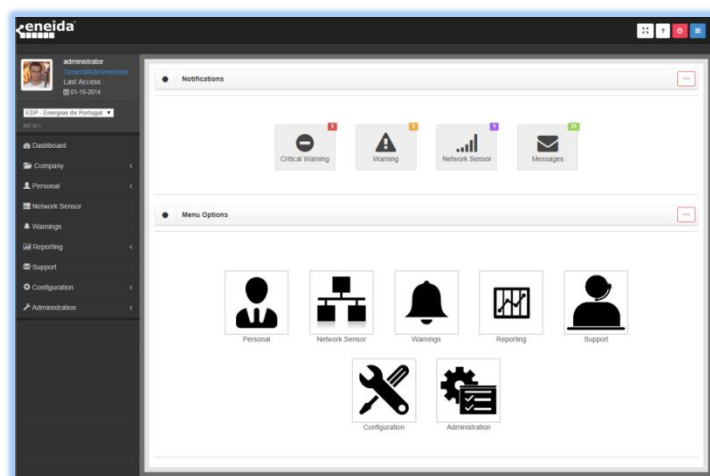


Figura 39 - Vista Principal da aplicação SCADA Web

Neste capítulo foram apresentadas as diferentes aplicações desenvolvidas sobre a plataforma *.NET* com recurso às diferentes tecnologias disponibilizadas por esta *framework*, para o novo sistema *SCADA* da *eneidaws*, tendo-se apresentado genericamente a forma como estas aplicações foram estruturadas e desenvolvidas. No Anexo E, apresentam-se os testes realizados a estas aplicações. No capítulo seguinte apresenta-se o caso de estudo da empresa e mostra-se de que forma este foi integrado no novo sistema desenvolvido.

8. CASO DE ESTUDO DA EMPRESA

Neste capítulo apresenta-se o caso de estudo que a *eneidaws* tem atualmente para um cliente seu. Na secção 8.1, é feito o enquadramento do caso de estudo, onde são explicadas as necessidades pretendidas pelo seu cliente, e é feita uma explicação do que é um posto de transformação e *hardware* produzido pela *eneidaws* para este caso de estudo. Na secção 8.2, demonstra-se como foi possível integrar este caso de estudo da empresa no projeto desenvolvido.

8.1. OBJETIVO DO CASO DE ESTUDO

O caso de estudo tem como objetivo poder proporcionar a integração no novo sistema *SCADA* desenvolvido e permitindo usar-se informação como dados reais de uma rede de sensores de um cliente real da *eneidaws*, possibilitando-se desta forma validar o resultado do trabalho desenvolvido. Para isso será apresentado o funcionamento do caso de estudo da empresa e posteriormente será demonstrado de que forma este caso de estudo real da *eneidaws* foi integrado no novo sistema *SCADA*.

8.2. ENQUADRAMENTO DO CASO DE ESTUDO

A *eneidaws* trata atualmente do caso de estudo que tem como finalidade a monitorização e diagnóstico dos postos de transformação (PT) para um cliente seu. O cliente da *eneidaws* tem como necessidade monitorizar os postos de transformação, a nível ambiental, energético, de segurança, qualidade do serviço, entre outros; pretende que a informação esteja centralizada, tenha a possibilidade de configurar regras para alarmes, aceder à aplicação remotamente em qualquer lugar e que o acesso à aplicação pelos utilizadores seja restrito.

A *eneidaws* tem atualmente *hardware* que constitui uma rede de sensores de teste, que se encontra instalada no posto de transformação (figura 40), para monitorização das condições e diagnóstico dos componentes elétricos existentes neste PT. A nível de *software*, a *eneidaws* usa atualmente aplicação servidor para recolha dos dados da rede de sensores e armazena os dados em base de dados, e a aplicação *SCADA DSI*, que apresenta várias limitações em funcionalidades, confirma a necessidade de um *software* que seja capaz de fornecer as funcionalidades pretendidas pelo cliente e facilite o tratamento e diagnóstico de dados das redes sensores instaladas nos diferentes postos de transformação.

Para ter-se uma ideia geral do que trata o caso de estudo, precisa-se saber o que é um PT, como poderá ver-se na secção 8.1.1, e o *hardware* produzido pela *eneidaws* usado neste caso de estudo que tem como finalidade adquirir dados do PT, visto na secção 8.1.2.

8.2.1. Posto de Transformação

O posto de transformação é uma instalação onde se procede à transformação da energia elétrica de média tensão para baixa tensão, alimentando a rede de distribuição de baixa tensão.

Um posto de transformação é constituído essencialmente pelos três componentes seguintes:

- **Equipamentos de interrupção ou seccionadores** – Estes permitem fazer cortes de energia das diferentes fases ou linhas (fase 1,2,3), para proteção, manutenção e intervenção técnica de um PT;
- **Transformadores** – Estes têm como responsabilidade fazer a transformação da energia de média tensão para baixa tensão.
- **Quadro geral de baixa tensão** – Depois de transformada a energia de média tensão para baixa tensão, as diferentes linhas ou fases 1,2,3, são distribuídas por diversos circuitos ou ramais da rede baixa tensão, que vão fornecer energia elétrica às empresas, casas, entre outros.

Na figura 40 apresenta-se um esquema genérico, com o objetivo de demonstrar de uma forma simples a ligação dos componentes no posto de transformação e de que forma é feita a transformação da energia de média tensão para baixa tensão. As setas na figura 40 representam o fluxo da energia elétrica de média tensão para baixa tensão.

Os postos de transformação podem ter duas tipologias, aéreos ou de cabine. Os postos de transformação aéreos têm o transformador instalado num apoio da linha de distribuição de média tensão, e o quadro geral de baixa tensão na base desse apoio, num armário dimensionado para o efeito.

Os postos de transformação com tipologia em cabine têm todos os equipamentos instalados dentro de uma cabine. Os PT desta tipologia cabine podem assumir uma das seguintes variantes: cabine alta (torre); cabine baixa num edifício próprio; cabine metálica (monobloco) ou cabine subterrânea (Wikipédia, 2013).

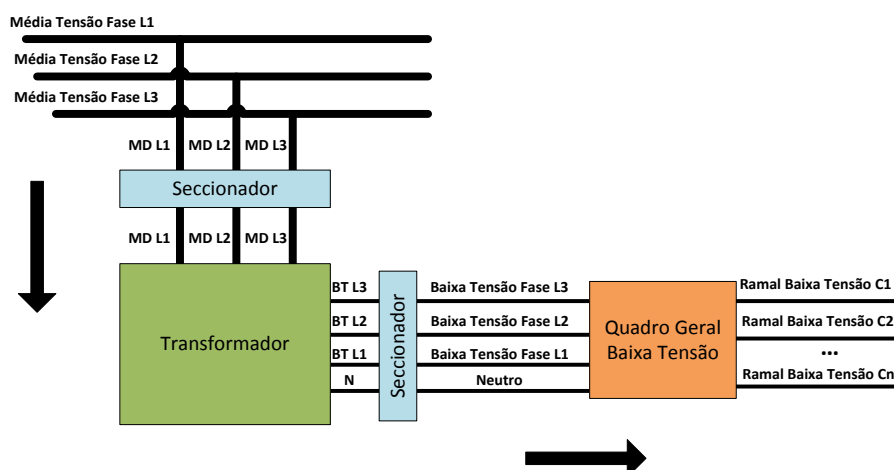


Figura 40 - Esquema simplificado de um Posto de Transformação

Na figura 41 apresenta-se o posto de transformação do tipo cabine, que está a ser usado pela *eneidaws* para realizar testes reais na aquisição de dados dos sensores instalados neste posto de transformação.



Figura 41 - Posto de Transformação com tipologia de cabine usado no caso de estudo pela *eneidaws*

8.2.2. Hardware produzido pela *eneidaws*

A *eneidaws* teve como base de ação decompor o caso de estudo em diferentes áreas de incidência, que representem benefícios na monitorização dos postos de transformação para o seu cliente. Neste sentido teve necessidade de desenvolver uma gama de *hardware*, apresentada nas figuras 42 e 43 a descrição do *hardware* desenvolvido pela *eneidaws*.

Cada *hardware* desenvolvido para as áreas de incidência definidas refletiu-se nos seguintes benefícios para o cliente da *eneidaws*:

- **Redução de Riscos Ambientais** – Este benefício é conseguido através dos sensores: *EWS DTHT*; *EWS DTH2O*; *EWS DTT*; *EWS DTPump*. Particularmente estes sensores têm como responsabilidade a redução de riscos ambientais, verificando a ocorrência de incêndios, inundações nos PT subterrâneos e avarias na ventilação.
- **Segurança e Controlo de Acessos** – Este benefício é conseguido através dos sensores *EWS DTP* e *EWS DTP_g*. Estes têm como responsabilidade a verificação do controlo de acessos, a deteção de intrusão, a remoção de componentes do PT.
- **Qualidade do Serviço** – Este benefício é conseguido através dos sensores: *EWS DTF*, que verifica o estado dos fusíveis no quadro geral de baixa tensão; *EWS DTTI_reh*, para monitorização da corrente das linhas 1,2,3 dos diferentes circuitos do quadro geral de baixa tensão; *EWS DTV*, para monitorização da tensão; e *EWS DTPD*, que serve para analisar as descargas parciais, verificando o nível de isolamento do PT. Estes sensores servem para garantir que a energia é fornecida aos clientes de acordo com os padrões dos reguladores do setor energético.
- **Perdas Técnicas e Não Técnicas** – Este benefício é conseguido através dos sensores: *EWS DTTI_reh*, para monitorização da corrente das linhas 1,2,3 dos diferentes circuitos do quadro geral de baixa tensão; *EWS DTV*, para monitorização da tensão. Com estes sensores pretende-se verificar os desequilíbrios da corrente e da tensão onde se pode verificar a ocorrência de perdas nas linhas 1,2,3. Se estas perdas se verificarem, a temperatura do transformador aumenta; para isso, usa-se o *EWS DTT*, responsável por analisar a temperatura de contacto num componente elétrico de um PT. O *EWS DTPD* serve para verificar as descargas parciais, analisando o nível de isolamento do PT. Estes sensores servem para garantir o funcionamento do posto de transformação e, na eventualidade de um problema, a sua rápida resolução.

É essencial referir que este caso de estudo é recente na *eneidaws*. Por esta razão, existem alguns sensores que não estão desenvolvidos, encontrando-se em fase de desenvolvimento, como é o caso do *EWS DTV_reh*, do *EWS DTH2O* e do *EWS DAT3T*.

Os restantes sensores desenvolvidos são protótipos que estão a ser validados numa fase experimental no terreno, para garantir que têm efetivamente o comportamento esperado na aquisição dos dados do PT.



Figura 42 - Descrição dos sensores desenvolvidos para monitorização dos postos de transformação e suas características, eneidaaws [documento interno da empresa]

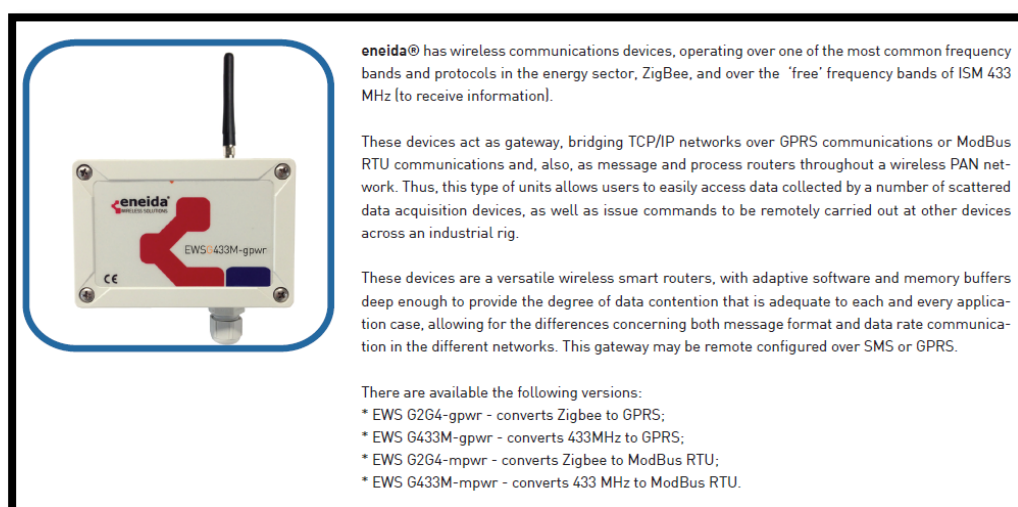


Figura 43 - Descrição da gateway desenvolvida para fazer a ponte entre a rede de sensores e a aplicação servidor e suas características, eneidaaws [documento interno da empresa]

8.2.3. Rede instalada no Posto de Transformação

O tipo de comunicação usada na rede de sensores do PT é sem fios. A comunicação usada opera sobre as frequências mais comuns, como protocolos *Zigbee* (Zigbee Alliance, 2014) e frequência ISM 433 MHz (Wikipédia, 2014b), para comunicação entre sensores e dos sensores com a *gateway*. A *gateway* funciona por *TCP/IP*, através da internet, por *GPRS* ou por outros meios físicos, com a aplicação servidor, onde são recolhidos os dados adquiridos do posto de transformação em base de dados.

Nas figuras 44, 45, 46, 47 e 48, apresenta-se a instalação dos sensores e da *gateway* no posto de transformação atual. Estas imagens referentes ao PT foram retiradas do documento interno da *eneidaws*.

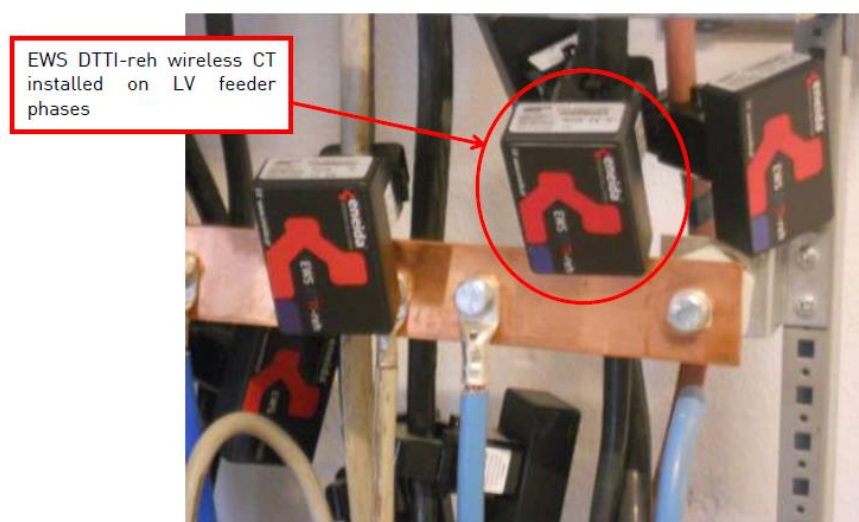


Figure 2 - EWS DTTI-reh wireless CT installed on LV feeder phases.

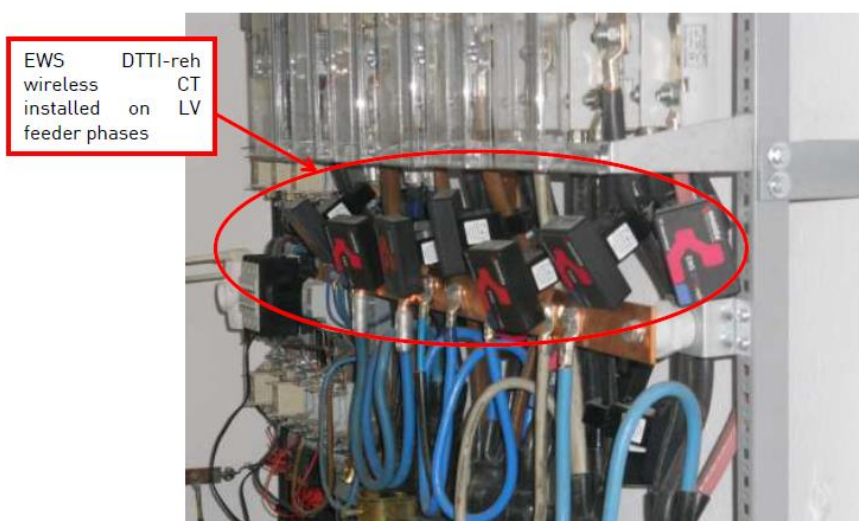


Figure 3 - Several EWS DTTI-reh wireless CT installed on LV feeder phases.

Figura 44 - Instalação dos sensores EWS DTTI_reh para medir a corrente em cada circuito do quadro geral do PT, *eneidaws* [documento interno]



Figure 8 – EWS DTH20-r wireless water detection sensor.

Figura 45 - Instalação do sensor EWS DTH20 para verificar a existência de inundação no PT, eneidaws [documento interno]



Figure 10 – EWS DTP-r wireless position sensor installed to detect the door opening.

Figura 46 - Instalação do sensor EWS DTP para controlar a abertura e fecho da porta no PT, eneidaws [documento interno]



Figure 11 – EWS DTHT-r installed on the ceiling of the secondary substation.

Figura 47 - Instalação do sensor EWS DTHT para medir temperatura e humidade no PT, eneidaws [documento interno]



Figure 12 – EWS DTPD-rpwr wireless and non-intrusive partial discharges sensor.

Figura 48 - Instalação do sensor EWS DTPD para medir as descargas parciais no PT, eneidaws [documento interno]

Na figura 49 apresenta-se o quadro geral de baixa tensão e a instalação dos sensores no PT. No quadrado vermelho nº 1 da figura 49 vê-se a *gateway EWS G433M*, responsável por reportar os dados adquiridos pelos sensores no PT para a aplicação servidor. No quadrado vermelho nº 3 da figura 49 vê-se o sensor *EWS DTPD*, responsável por medir as descargas parciais no PT. No retângulo vermelho nº 5 da figura 49 veem-se os sensores *EWS DTTI_reh*, responsáveis por medir a corrente em cada circuito, e *EWS DTF*, responsáveis por verificar o estado dos fusíveis dos vários circuitos (C1, C2, C3, C4, C5, C6) do quadro geral visto no quadrado nº 4 da figura 49. No quadrado vermelho nº 2 da figura 49 vê-se o seccionador que

permite fazer cortes de energia das diferentes fases ou linhas (fase L1, L2, L3), para proteção e intervenção técnica no PT.

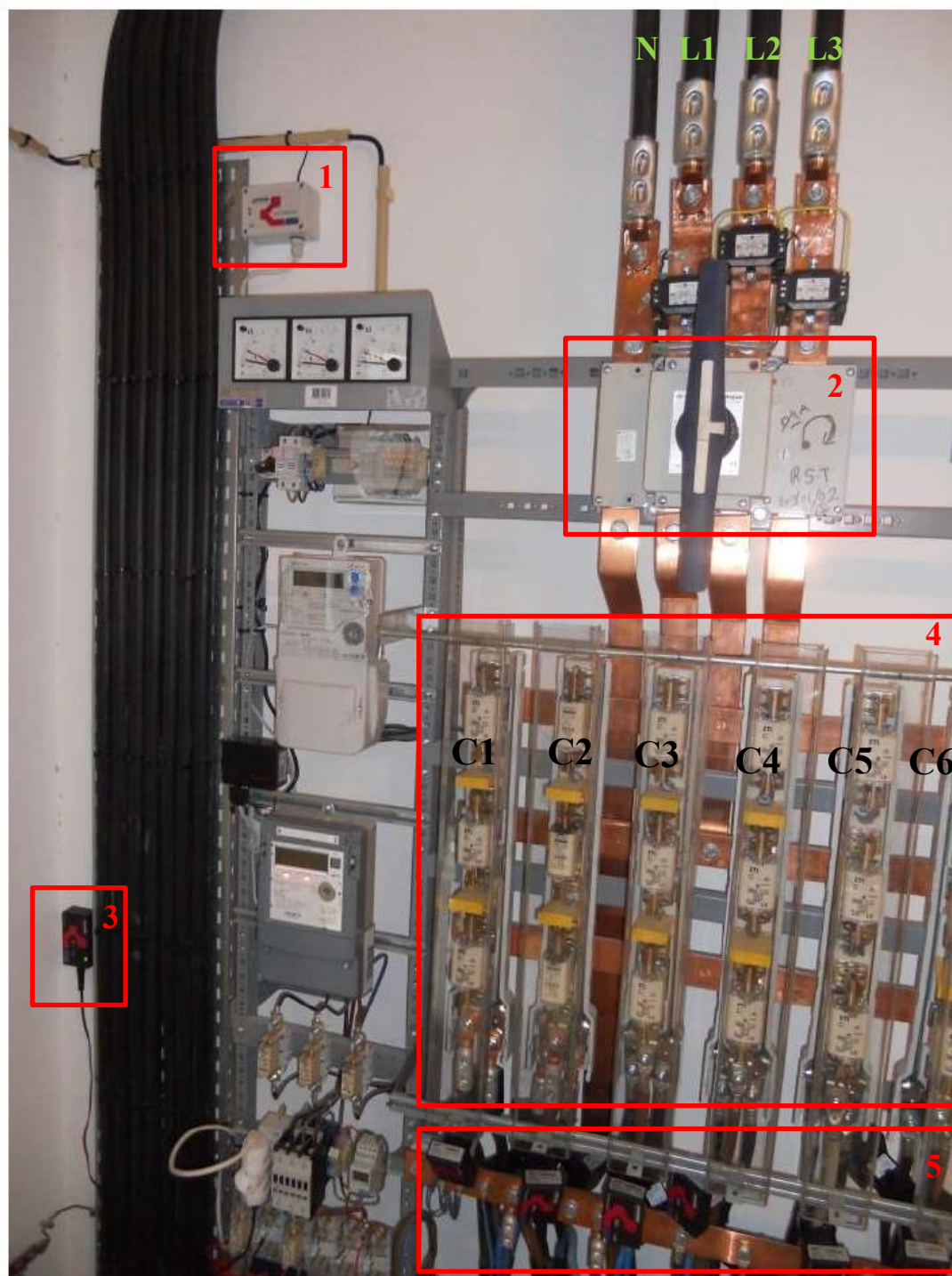


Figure 15 – General view of the installed sensors.

Figura 49 - Visão geral da instalação dos sensores no PT, eneidaws [documento interno]

8.2.4. Aplicação SCADA DS1 usada no caso de estudo

No capítulo 2, foram apresentadas algumas limitações da aplicação *SCADA DS1* (inexistência de controlo de acessos, configuração de regras e ações para alarmes, automatismos para importação de redes de sensores, uma área para pesquisa customizada de informação pelo utilizador, entre outros). Este *software* é usado pela *eneidaws* nos seus diferentes projetos e também neste caso de estudo. Pode igualmente constatar-se a carência de funcionalidades e opções que se verifica nesta aplicação.

Concretamente para este caso de estudo, esta aplicação só permite a configuração da rede de sensores manualmente e a visualização do seu histórico de dados adquiridos da respetiva rede de sensores através de gráficos e tabelas. Na figura 50, apresenta-se o aspeto que a aplicação tem, podendo visualizar-se a árvore que representa a rede de sensores no PT, vista no lado esquerdo da figura 50, onde são apresentados os diferentes sensores. Selecionado cada um destes sensores, é possível visualizar a informação dos dados adquiridos entre uma data inicial e final em gráficos ou tabelas, não existindo mais funcionalidades e opções nesta aplicação. Verifica-se a inexistência de possibilidade de controlo de acessos na aplicação, visualização e configuração de regras para alarmes, pesquisas customizadas de informações, entre outros.

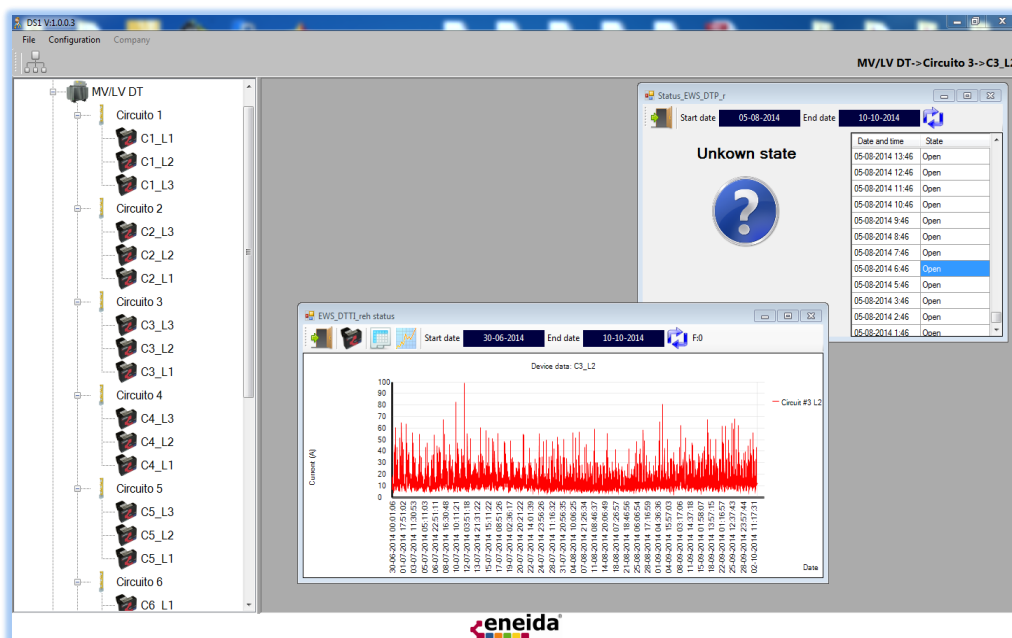


Figura 50 - Software SCADA DS1 - eneidaws [aplicação proprietária]

8.3. INTEGRAÇÃO DO CASO DE ESTUDO NO PROJETO DESENVOLVIDO

Nesta secção apresenta-se de forma esquemática a rede de sensores vista na figura 51 que se encontra instalada atualmente no PT, para testes dos diferentes sensores. Esta rede é constituída por: dezoito *EWS DTTI_reh*, sensores responsáveis por adquirir dados da corrente dos diferentes circuitos do quadro geral de baixa tensão; um *EWS DTF*, sensor responsável por verificar o estado dos fusíveis do quadro geral de baixa tensão; dois *EWS DTHT*, sensores responsáveis por ler temperatura e humidade do PT; um *EWS DTPD*, sensor responsável por verificar a existência de descargas parciais, indicando o nível de isolamento do PT; e um *EWS DTP*, sensor responsável por verificar a abertura e fecho da porta do PT em questão. Todos os sensores comunicam dentro do PT sem fios com a *gateway*, para esta posteriormente comunicar a informação para a aplicação servidor. Na representação do esquema visto na figura 51, recorreu-se à ligação dos sensores através de linhas para ajudar na percepção do fluxo de dados dos sensores com a *gateway*.

8.3.1. Cenário Real dos Sensores Instalados no PT

Na figura 51 apresenta-se o cenário real da rede de sensores existente atualmente no posto de transformação para testes.

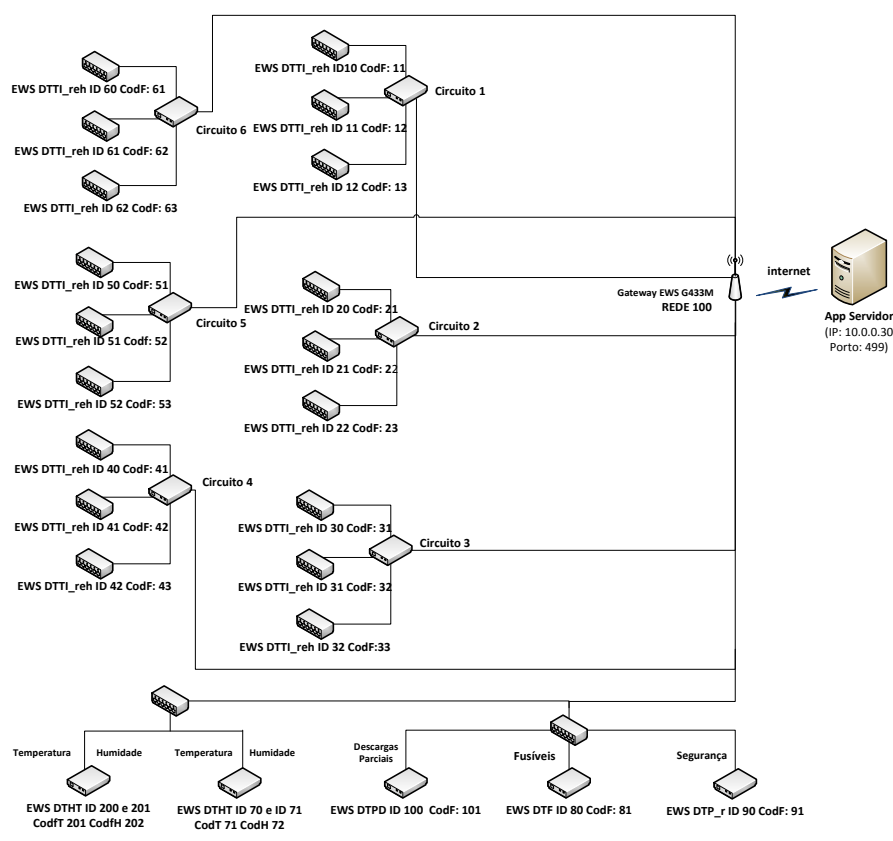


Figura 51 - Esquema representativo do cenário real da rede de sensores instalada no posto de transformação

8.3.2. Integração do Cenário na Aplicação SCADA Web

Nesta secção, apresentam-se algumas figuras com o resultado da integração deste caso de estudo na aplicação *SCADA Web* desenvolvida. Aqui constam todas as necessidades do cliente da *eneidaws*, para além de outras opções complementares às suas necessidades. Como os requisitos definidos para o novo sistema *SCADA* da *eneidaws* contemplam um maior leque de opções e funcionalidades, esse facto permite que este novo sistema *SCADA* seja versátil para a utilização nos mais diversos projetos que a *eneidaws* venha a ter.

Nesta aplicação, o cliente da *eneidaws* pode criar, editar e apagar utilizadores com diferentes níveis de privilégios. Cada utilizador que pretenda aceder à aplicação tem de fazer *login*, como se pode ver na figura 52; mediante os privilégios de acesso, o utilizador tem mais ou menos funcionalidades disponíveis. Os níveis de acesso de cada utilizador estão definidos com base nas regras de acesso à aplicação (*ClienteAdministrador* e *ClienteUtilizador*). Na figura 53, é apresentado o *dashboard* das funcionalidades de cada utilizador de forma dinâmica (por exemplo, o *ClienteAdministrador* pode criar utilizadores, redes de sensores, etc., e o *ClienteUtilizador* não tem acesso a estas opções por ter baixos níveis de privilégios).

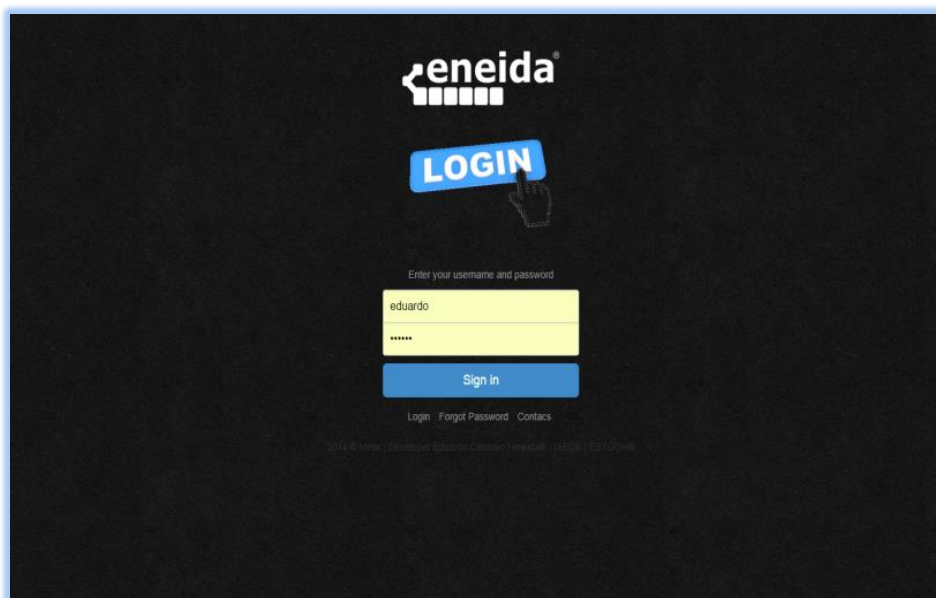


Figura 52 - Exemplo Aplicação SCADA Web Página de Login

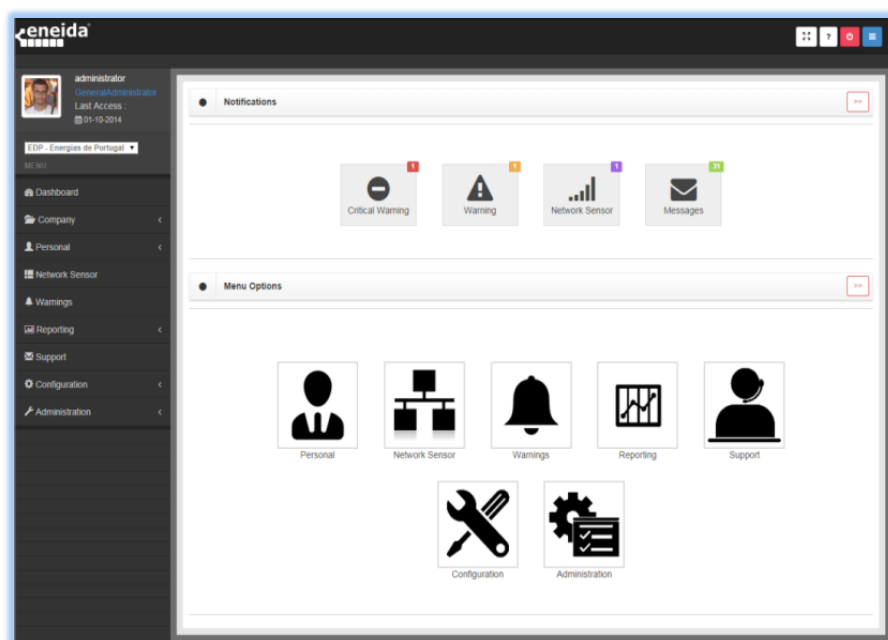


Figura 53 - Exemplo Aplicação SCADA Web Página Principal

O cliente da *eneidaws* pode pesquisar informação das redes de sensores entre uma data inicial e uma data final e analisar dados em tabelas ou gráficos, como se pode ver na figura 54. O utilizador tem ainda a possibilidade de exportar os dados para ficheiro Excel e exportar o gráfico visto para imagem, *JPG*, *PDF*, entre outros. No gráfico da figura 54 pode ver-se o comportamento de um sensor a nível temporal analisando quais foram os seus valores mínimos, médios e máximos na linha temporal pretendida e tirar as conclusões necessárias de acordo com o contexto.

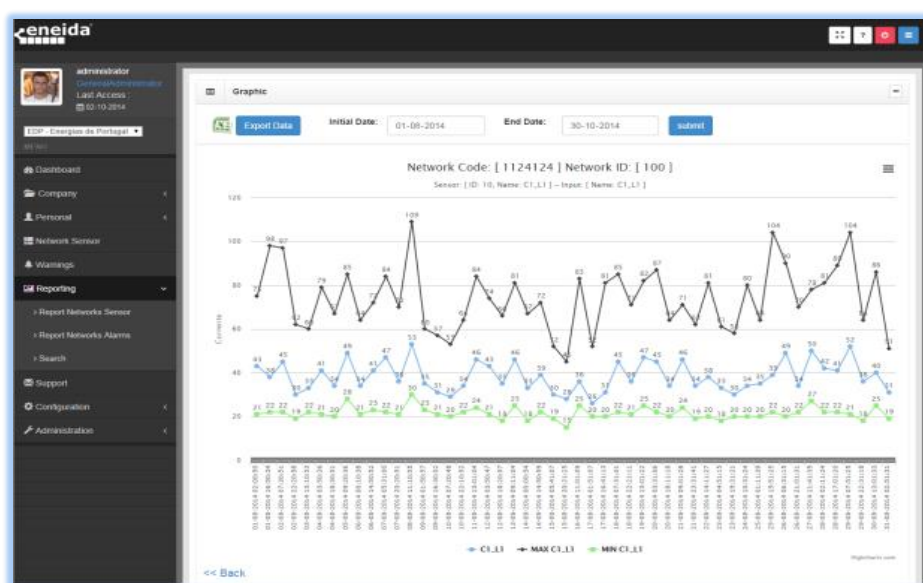


Figura 54 - Exemplo de página de reporte onde se vê um gráfico dos dados de um sensor na rede do PT do caso de estudo

Na figura 55, o cliente da *eneidaws* tem a possibilidade de ver as suas redes de sensores através de um mapa e pesquisar pelo código único definido para cada rede de sensores, ou, se preferir, pode clicar nos botões vermelho, amarelo e verde para visualizar só alarmes críticos, alerta de alarmes ou redes sem alarmes. O botão azul serve para visualizar todas as redes independentemente do seu estado (em alarme ou não). Como se pode ver na figura 55, sempre que o utilizador clicar num ponto pode ver informação resumida sobre a rede de sensores; caso o pretenda, pode ver a vista da rede de sensores clicando no link “+ info network”. Ao clicar neste link, o utilizador será redirecionado para a vista da rede de sensores com os últimos valores adquiridos nessa rede, como se pode ver na figura 56. A rede escolhida para este exemplo tem um alerta para alarme de cor amarela: pode-se ver em detalhe, na figura 55, que o problema está no sensor que obtém dados do circuito 1 e linha R.

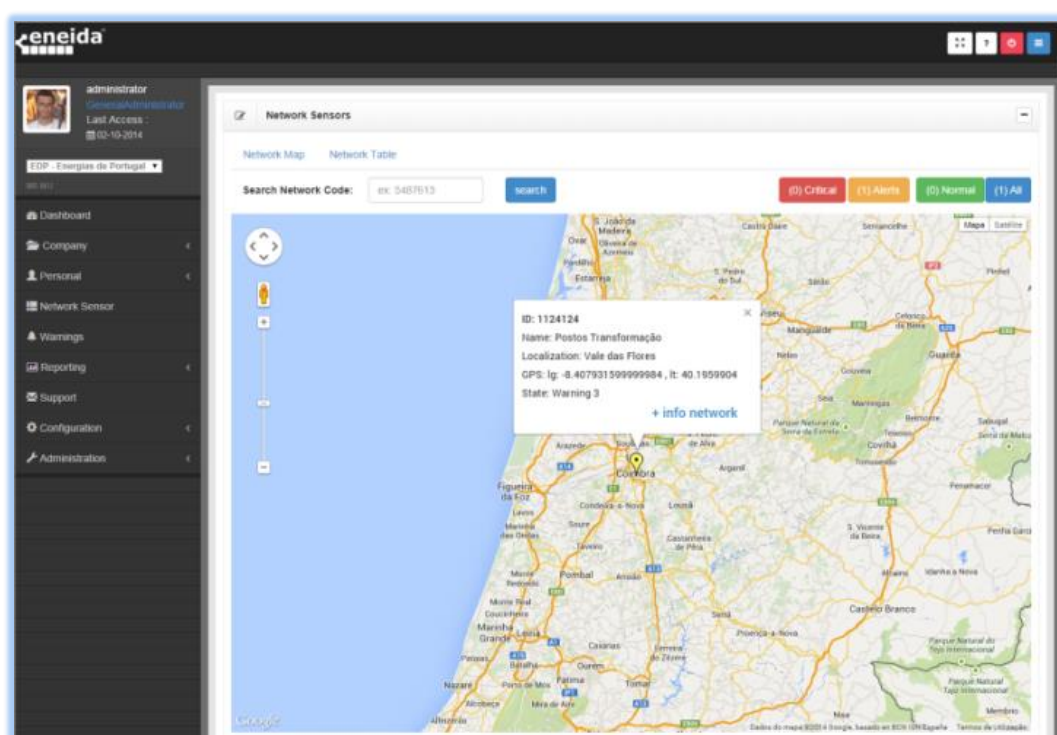


Figura 55 - Exemplo de página Aplicação SCADA Web Mapa Rede de Sensores

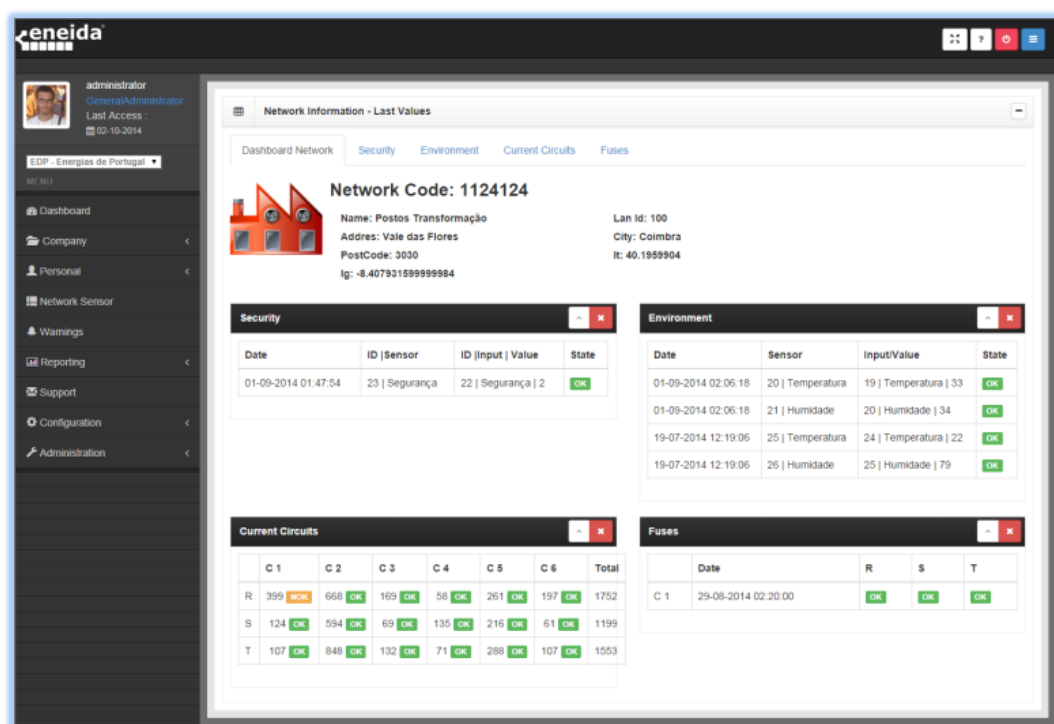


Figura 56 - Visualização dos últimos dados adquiridos na rede de sensor do PT do caso de estudo e estado de cada sensor

No Anexo D apresenta-se o manual de utilizador das aplicações: aqui podem ser vistas com mais detalhe as funcionalidades e opções que o cliente da *eneidaws* tem ao seu dispor para monitorização das redes de sensores dos PT, como, por exemplo, a importação automática de redes de sensores, a configuração de regras para alarmes, as pesquisas customizadas de informação, entre outras.

Neste capítulo, apresentou-se o caso de estudo que a *eneidaws* tem atualmente relativo aos postos de transformação, tendo sido explicadas as especificidades deste caso de estudo referente ao que é um posto de transformação e qual o seu funcionamento. As necessidades pretendidas pelo cliente da *eneidaws* neste caso de estudo acabam por ser satisfeitas pelas funcionalidades gerais do novo sistema *SCADA*, capaz de adaptar-se a um leque maior das necessidades existentes nos clientes da *eneidaws*. Apresentou-se ainda os vários equipamentos produzidos pela *eneidaws* para este caso de estudo, tendo sido possível explicar o que cada um representa e os benefícios obtidos através dos mesmos. Foi também apresentado o resultado da integração deste caso de estudo no projeto desenvolvido. No capítulo 9, será feita uma análise crítica de todo o trabalho desenvolvido, onde será possível demonstrar as vantagens gerais do desenvolvimento deste projeto para a *eneidaws* e em específico para este caso de estudo.

9. ANÁLISE CRÍTICA

Neste capítulo apresenta-se a análise crítica do novo sistema *SCADA* desenvolvido em virtude das necessidades e expectativas da empresa e das perspectivas do aluno na proposta para implementação do novo sistema, presentes na secção 9.1. Na secção 9.2, apresentam-se alguns cenários mais prováveis em que o novo sistema *SCADA* pode ser usado pela empresa, com o objetivo de reforçar a flexibilidade deste novo sistema *SCADA*.

9.1. NOVO SISTEMA *SCADA*

O resultado final do desenvolvimento do novo sistema é bastante positivo, pois se analisarmos o sistema atualmente em uso pela *eneidaws* e o novo sistema desenvolvido neste projeto para a empresa prevê-se que este trará valor acrescentado à *eneidaws* por apresentar mais funcionalidades e opções gerais que se prevê irem ao encontro das necessidades dos clientes da *eneidaws*. Os resultados positivos dos testes presentes no Anexo E vêm, de igual forma, comprovar o desfecho final positivo.

Todos os requisitos assinalados pela empresa e propostos pelo aluno foram contemplados no desenvolvimento e na adição de mais funcionalidades ao novo sistema *SCADA*, com vista a satisfazer as necessidades da *eneidaws*. O desenvolvimento do novo sistema está assente em novas arquiteturas de desenvolvimento para as suas aplicações, que permitiram a integração de novas funcionalidades e opções no futuro. Foram usadas novas tecnologias *Windows*, como é o caso da aplicação *SCADA Web*, enriquecida com um maior leque de funcionalidades e opções para supervisão e consulta das redes de sensores, que permitirá o funcionamento em diferentes dispositivos, bastando o recurso a um *browser*; uma nova aplicação servidor para a receção de mensagens com a capacidade de verificar existência de alarmes; uma nova aplicação baseada em *Web Service*, que permitirá uma nova forma de disponibilização de dados e integração com outros sistemas; e uma nova aplicação serviço de alarmes para verificação de alarmes e notificação do utilizador por *e-mail* e *SMS*. A aplicação *Web Service* e a aplicação serviço de alarmes não existiam no sistema atual da *eneidaws*, não sendo possível fazer uma comparação com o sistema atual.

No entanto, através da tabela 25 e da tabela 26, fez-se uma análise comparativa relativamente a funcionalidades e opções que a aplicação servidor (atual e nova), e a aplicação *SCADA* (atual e nova) têm, com a qual se confirmam as mais-valias no novo sistema *SCADA* desenvolvido para a *eneidaws*. Como foi visto no capítulo 8, este novo sistema *SCADA* permitiu a integração do caso de estudo da *eneidaws*. Esta situação verificou-se por este sistema ter sido pensado para funcionar em diferentes contextos.

Tabela 25 - Comparativo Aplicação SCADA Atual e Nova Aplicação SCADA Web

Funcionalidades	Aplicação Atual SCADA DS1	Nova Aplicação SCADA Web
Controlo de Acessos na aplicação.		
Gestão de Empresas (Criar, Editar, Adicionar, Remover).		
Gestão de Utilizadores (Criar, Editar, Adicionar, Remover).		
Gestão de Licenças (Criar, Editar, Adicionar, Remover).		
Criar Rede Sensores (Criar, Editar, Adicionar, Remover).		
Importação e Exportação Automática de Rede de Sensores.		
Visualização das Redes de sensores através de coordenadas GPS, num mapa ou tabela.		
Dashboard para visualizar os últimos dados adquiridos por uma rede de sensores e saber o estado de cada sensor (em alarme ou não).		
Gestão de alarmes ocorridos, com possibilidade de adicionar observações ao alarme para histórico.		
Gestão de Regras e Ações para alarmes nas redes de sensores (Criar, Editar, Adicionar, Remover).		
Suporte aos utilizadores da aplicação por parte do administrador.		
Visualização de gráficos e tabelas, histórico de dados.		
Visualização de gráficos e tabelas e exportação desses dados para Excel, PDF, etc.		
Pesquisa customizada de informação relativa a uma rede de sensores (por exemplo, saber últimos sensores que geraram alarmes, sensores que não comunicam há mais de um determinado tempo, ver últimos valores adquiridos por um sensor de uma rede, possibilidade de exportar essa informação para Excel, etc).		
Notificação do utilizador quando existem erros nas configurações relativas a uma determinada rede de sensores.		
A aplicação funciona em diferentes dispositivos e diferentes sistemas operativos.		

Tabela 26 - Comparativo Aplicação Servidor Atual e da Nova Aplicação Servidor

Funcionalidades	Aplicação Servidor Atual	Nova Aplicação Servidor
Capacidade de comunicar com a rede de sensores através da estrutura de mensagens definida nas comunicações pela <i>eneidaws</i> .	✓	✓
Verificação de alarmes na existência de um alarme numa mensagem recebida por uma rede de sensores.	✗	✓
Possibilidade de parametrizar a aplicação servidor para ser capaz de enviar <i>e-mail</i> e <i>SMS</i> ao utilizador na ocorrência de um alarme.	✗	✓
Aplicação servidor opera localmente num sistema <i>Windows</i> , funcionando em <i>background</i> , após o sistema estar parametrizado.	✓	✓
Capacidade de identificar os dados adquiridos das redes de sensores e guardar esses dados na base de dados de cada cliente.	✗	✓
Possibilidade de fazer <i>debug</i> das mensagens recebidas através do <i>log</i> na aplicação servidor.	✓	✓

9.2. CENÁRIOS DA APLICAÇÃO DO NOVO SISTEMA SCADA

Nesta secção apresentam-se alguns cenários mais prováveis em que a empresa *eneidaws* possa vir a empregar as aplicações, apesar de a aplicabilidade das mesmas poder ser feita em outros contextos. Desta forma, demonstra-se como o novo sistema é flexível e modelar para utilizar em diferentes cenários.

Para o novo sistema funcionar nos cenários apresentados onde são configuradas as aplicações, exigem-se os seguintes requisitos mínimos:

- Um computador com sistema operativo *Windows 7*, ou, preferencialmente, um servidor instalado com *Windows Server 2008*, para alojar e executar as quatro aplicações;
- O servidor *Web Internet Information Server (IIS)* deve estar instalado para alojar a aplicação *Web Service* e a aplicação *SCADA Web*. É importante ter em conta que, se existirem mais de 1000 utilizadores a usar em simultâneo a aplicação *Web Service* ou aplicação *SCADA Web*, o sistema operativo a ser usado terá de ser uma versão *Windows Server*, pois o servidor *IIS* não tem limites no número de acessos em A validação de alarmes na componente aplicacional e o armazenamento de dados das redes de sensores, deve ser usada a aplicação servidor para guardar dados com a

simultâneo. Na versão *Windows 7*, o *IIS* tem um limite de acessos em simultâneo, ideal para usar em sistemas muito pequenos, por exemplo numa *intranet*;

- Para validação de alarmes na componente aplicacional e o armazenamento de dados das redes de sensores, deve ser usada a aplicação servidor para guardar dados com a opção validar alarmes ativa, e a aplicação *SCADA Web* para monitorização, diagnóstico e controlo dos dados adquiridos;
- Deve ter-se instalado o motor de base de dados *Microsoft SQL Server Express* versão gratuita, com limite máximo de 10 GB por base de dados, ou *Microsoft SQL Server* versão completa, paga, sem limite no tamanho da base de dados, para guardar os dados das aplicações e das redes de sensores. Numa fase inicial pode ser usada a versão *Express* e, posteriormente, passar-se para uma versão completa;
- Para existir comunicação entre a rede de sensores e as aplicações, será necessária uma ligação à internet por parte da rede de sensores e uma outra ligação por parte das aplicações (o tipo de ligação à internet dependerá sempre do meio físico em que o sistema está a ser projetado; poderá não ser necessária uma ligação à internet se todo o sistema estiver dentro de uma rede local);
- É necessário um domínio *Domain Name Server (DNS)* que permita à rede de sensores, através da internet, conectar-se à aplicação servidor. Este *DNS* deve ser configurado no *router* que permite o acesso local à aplicação servidor. Neste sentido, o *DNS* permite fazer tradução nome para *IP* público e o *router* de *IP* público para *IP* privado, permitindo o acesso ao servidor.

9.2.1. Cenário 1 Online – Empresa de Pequena Dimensão

Na figura 57, apresenta-se a visão simplificada do novo sistema que poderá ser usado na maior parte dos casos em que a *eneidaws* projete redes de sensores, para empresas de pequena dimensão que pretendam usar o novo sistema através da internet.

As redes de sensores adquirem dados e submetem-nos para a aplicação servidor. A aplicação servidor valida a existência de alarmes, caso tenha a opção ativa para verificar alarmes e notificar o utilizador por *e-mail* ou *SMS*; caso contrário, guarda os dados adquiridos pela rede na base de dados do cliente respetivo.

A aplicação serviço de alarmes verifica a ocorrência de alarmes em todas as comunicações e comunica por *e-mail* ou *SMS* ao utilizador no caso de existência de alarmes.

A aplicação *Web Service* disponibiliza informações das redes de sensores no formato *JSON* através da internet por pedidos *HTTP* aos utilizadores ou a outros sistemas.

A aplicação *SCADA Web* permite ao utilizador da empresa fazer configurações, consultas, análises e diagnósticos de todas as informações referentes às suas redes de sensores através de um *browser* em qualquer local.

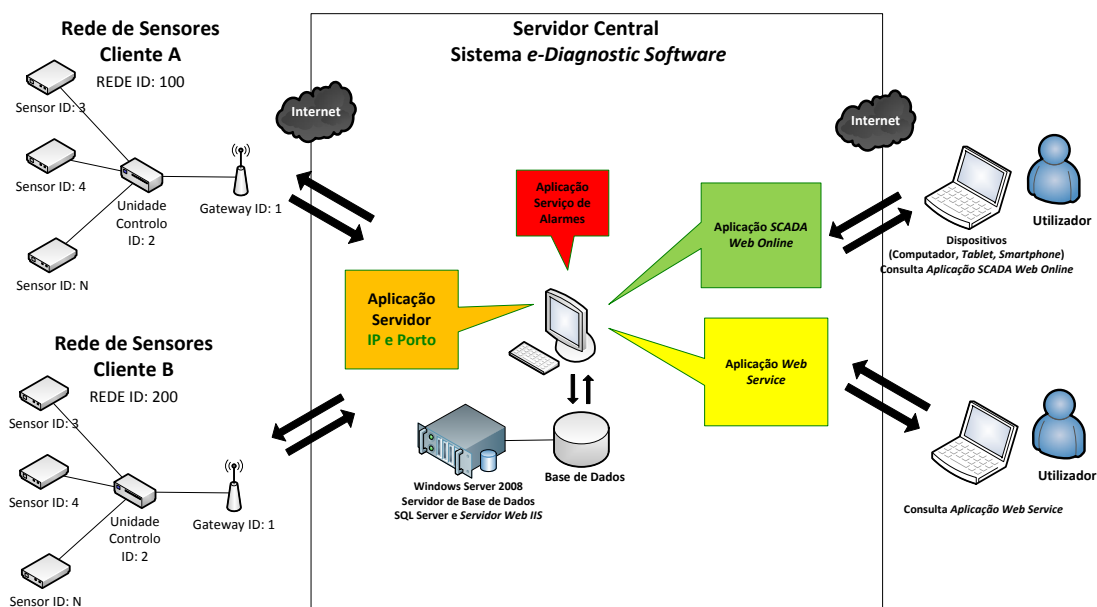


Figura 57 - Cenário 1 geral do sistema através da internet

9.2.2. Cenário 2 Online – Empresa de Grande Dimensão

Na figura 58, é exemplificado um cenário em que existe uma grande quantidade de dados gerada pela rede de sensores de um cliente A e a rede de sensores de um cliente B, ambos empresas de grande dimensão. Nesta situação, em que cada uma das redes dos clientes tem uma grande quantidade de sensores a adquirir dados, pode ser usado um servidor dedicado a adquirir os dados da rede de cada um dos clientes para balanceamento de carga, sendo que cada uma das aplicações servidor guarda a informação na respetiva base de dados do cliente A e do cliente B.

Nesta situação, é definido o mesmo *IP* para as aplicações servidor e um porto de comunicação diferente para cada rede de sensores. Esta situação verifica-se porque estão a ser usadas as duas aplicações servidor na mesma máquina física. Por sua vez, as redes de sensores, ao terem o *IP* e porto configurados, sabem com que aplicação servidor devem conectar-se e comunicar os dados das suas redes de sensores.

O funcionamento da aplicação serviço de alarmes, da aplicação *Web Service* e da aplicação *SCADA Web* é igual ao explicitado no cenário 1.

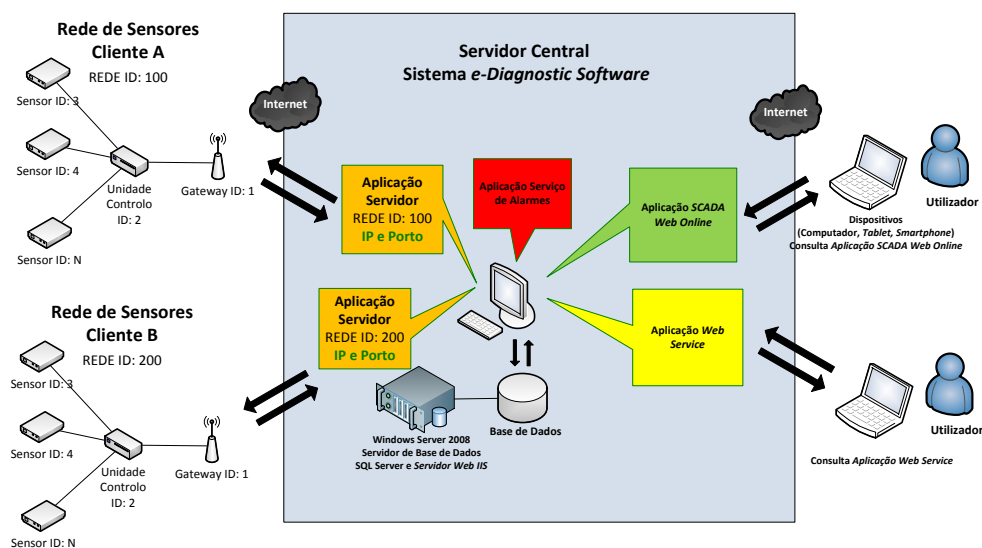


Figura 58 - Cenário 2 geral do sistema através da internet, com aplicações servidores dedicadas às redes de sensores de cada cliente

9.2.3. Cenário 3 Intranet – Empresa de Pequena Dimensão

Na figura 59, apresenta-se uma visão simplificada do novo sistema, demonstrando-se como o sistema funciona numa rede local *intranet* do cliente da *eneidaws*, em que a empresa é de pequena dimensão. De igual modo ao ambiente *online* na internet, cada rede de sensores adquire dados e submete-os para a aplicação servidor. Esta aplicação valida a existência de alarmes caso tenha a opção ativa para verificar alarmes e notificar o utilizador por *e-mail* ou *SMS*; caso contrário, guarda os dados adquiridos pela rede na base de dados.

O funcionamento da aplicação serviço de alarmes, da aplicação *Web Service* e da aplicação *SCADA Web* é igual ao explicitado no cenário 1.

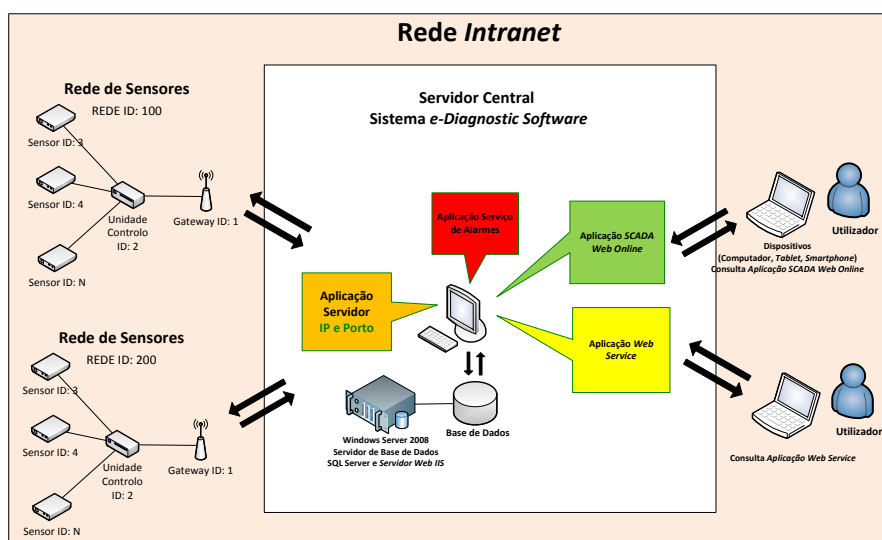


Figura 59 - Cenário 3 geral do sistema instalado numa rede local intranet de um cliente

9.2.4. Cenário 4 Intranet – Empresa de Grande Dimensão

Na figura 60, apresenta-se o exemplo de quando existe uma grande quantidade de dados gerada por uma rede de sensores *id* 100 e uma rede de sensores *id* 200, num ambiente *intranet*, rede local de um cliente, em que a empresa é de grande dimensão. Nesta situação pode ser usado um servidor dedicado a adquirir os dados de cada uma das redes para balanceamento de carga, cenário semelhante ao cenário 2, com a diferença de este sistema estar a funcionar exclusivamente no ambiente *intranet* do cliente.

Nesta situação, é definido o mesmo *IP* para os servidores e um porto de comunicação diferente para cada rede de sensores. Esta situação verifica-se porque estão a ser usadas as duas aplicações servidor na mesma máquina física. Por sua vez, as redes de sensores, ao terem o *IP* e porto configurados, sabem com que servidor devem conectar-se e comunicar os seus dados.

O funcionamento da aplicação serviço de alarmes, da aplicação *Web Service* e da aplicação *SCADA Web* é igual ao explicitado no cenário 1.

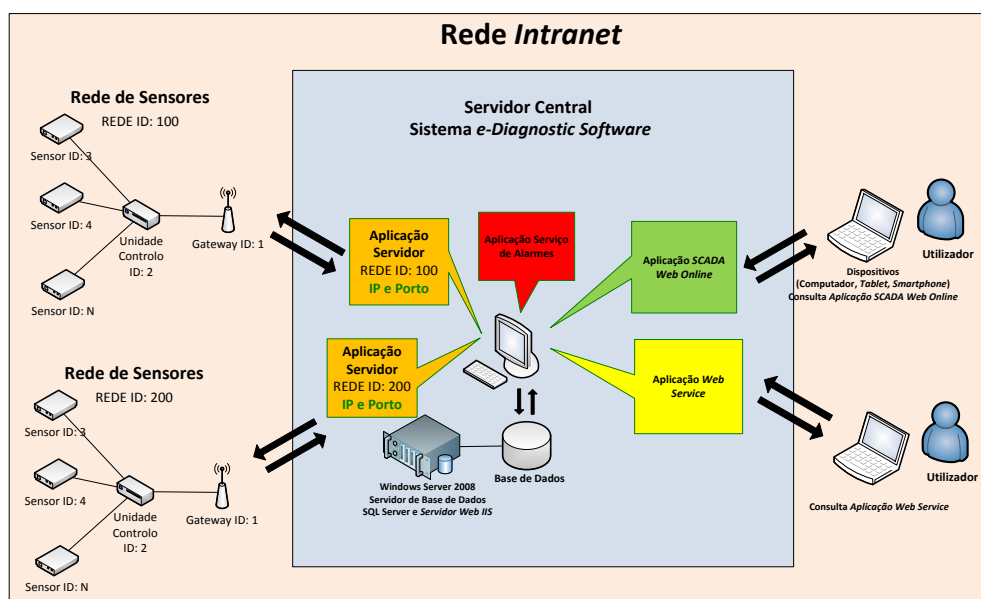


Figura 60 - Cenário 4 geral do sistema instalado numa rede local intranet de um cliente, com aplicações servidores dedicadas para cada uma das redes de sensores instaladas

Foram apresentados quatro cenários que a *eneidaws* possa vir a encontrar e em que aplique o novo sistema *SCADA*. No entanto, este novo sistema *SCADA* poderá ser aplicado em diferentes tipos de infraestruturas físicas para comunicação de dados das redes de sensores ou uso de várias máquinas físicas para alojar as aplicações para, por exemplo, balanceamento de carga e performance. Desenvolveu-se este sistema para a *eneidaws* com o objetivo de ser tão flexível quanto possível para a estruturação de novas soluções e integração dos projetos neste sistema.

10. CONCLUSÕES E TRABALHOS FUTUROS

Com a elaboração do presente projeto, pretendia-se melhorar o atual sistema *SCADA* para supervisão e diagnóstico das redes de sensores da *eneidaws*, criando um novo sistema *SCADA*. Para isso, foi necessário, numa fase inicial, perceber as necessidades principais que a *eneidaws* tinha no atual sistema e, para além disso, analisar as limitações que o sistema apresentava. Numa segunda fase, analisou-se os sistemas *SCADA* existentes no mercado, para perceber quais as funcionalidades base principais existentes neste género de aplicações, e fez-se um comparativo com o sistema atual da *eneidaws*, onde se comprovou as limitações a nível de funcionalidades e opções. Conhecendo o aluno as limitações do sistema *SCADA* da *eneidaws* e os requisitos iniciais pretendidos pela empresa, e tendo em conta as principais funcionalidades base existentes neste género aplicações, elaborou uma proposta de novas aplicações contemplando mais funcionalidades e opções à *eneidaws* para o desenvolvimento do novo sistema *SCADA*. Na fase final de refinamento de toda a informação necessária para desenvolvimento deste projeto, o aluno apresentou uma solução para o desenvolvimento do projeto e a proposta de requisitos necessários previamente validados pela *eneidaws*, onde se contemplava o desenvolvimento de quatro aplicações (uma nova aplicação servidor, uma aplicação serviço de alarmes, um aplicação baseada em *Web Service* e uma nova aplicação *SCADA Web*). O novo sistema teria que apresentar as seguintes características: uma arquitetura modular e flexível que facilitasse a integração de novas e futuras funcionalidades, contemplasse o uso de novas tecnologias multiplataforma capazes de funcionar com diferentes dispositivos e apresentasse a informação centralizada.

Especificamente, conseguiu-se alcançar os seguintes objetivos: implementação de um novo modelo de dados capaz de albergar o novo sistema *SCADA*; desenvolvimento de uma nova aplicação servidor capaz de desempenhar mais funções, opcionalmente a validação e tratamento de alarmes capazes de enviar notificações para o utilizador por *e-mail*, *SMS*, e a separação dos dados das redes sensores pelas diferentes bases de dados dos clientes; uma nova aplicação serviço de alarmes com a função exclusiva de tratamento de alarmes e envio de notificações para o utilizador por *e-mail* e *SMS*; uma nova aplicação baseada em *Web Service*, que possibilita a disponibilização de informação das redes de sensores aos clientes da *eneidaws*, viabilizando uma arquitetura distribuída, que permita integrar a informação com outras aplicações, independentemente do tipo de tecnologia usada nessas aplicações, bastando que estas tenham a capacidade de fazer pedidos por *HTTP* ao *Web Service*.

O aspeto principal do desenvolvimento deste projeto centrou-se na criação de uma nova aplicação *SCADA Web* para supervisão e análise das redes de sensores, capaz de funcionar em diferentes sistemas operativos e dispositivos, bastando o recurso a um *browser*. Esta aplicação *SCADA Web* está otimizada com uma gama nova de funcionalidades e opções: o recurso a regras de acesso à aplicação com o controlo de acessos através de contas de utilizadores e

diferentes níveis de privilégios; uma área de alarmes para criar regras para alarmes e gestão dos mesmos; uma área para tratamento e análise de dados adquiridos das redes de sensores através de gráficos, tabelas e exportação destes dados para Excel, entre outros formatos; a configuração manual de redes de sensores e a configuração automática de redes de sensores por importação de um ficheiro CSV com uma lista de redes de sensores; a exportação das redes de sensores para um ficheiro CSV; a visualização dos últimos dados adquiridos por uma rede de sensores; a visualização das redes de sensores por georreferenciação, entre outras funcionalidades.

Por fim, como objetivo final, conseguiu-se a integração do caso de estudo real da *eneidaws* referente aos postos de transformação no novo sistema SCADA, tendo sido usados dados reais da rede de sensores instalada no posto de transformação. Constatou-se o sucesso da aplicação implementada, ao contemplar uma série de funcionalidades genéricas capazes de apresentarem todas as funcionalidades que a empresa pretendia para este caso de estudo em específico e com base nos testes efetuados às aplicações desenvolvidas. Através do comparativo entre o sistema atual da empresa e o novo sistema desenvolvido neste projeto, prevê-se que este sistema trará valor acrescentado para a *eneidaws*, dada a existência de mais funcionalidades e opções, permitindo à empresa um leque maior de ofertas nas suas soluções que satisfaçam as necessidades dos seus atuais e potenciais clientes.

No futuro, será interessante o desenvolvimento de aplicações SCADA nativas para dispositivos móveis (como, por exemplo, *Android*, *Iphone*, *Windows Phone*, entre outros), nos quais estas podem obter informação sobre as redes de sensores através da aplicação desenvolvida baseada em *Web Service*. Estas aplicações poderão, assim, ser vistas como um complemento à aplicação SCADA Web. Será igualmente interessante criar um componente para a aplicação SCADA Web, que permita ao utilizador parametrizar informação sobre as redes de sensores, para este receber relatórios automaticamente por *e-mail*. O envio poderá ser feito com a periodicidade que o utilizador pretenda (diariamente, semanalmente ou mensalmente). O *design* gráfico da aplicação SCADA Web poderá ser otimizado, visto o aluno não ser especialista em *design*. Seria igualmente interessante criar um componente no qual pudessem ser criadas várias animações gráficas relativamente aos diferentes *dashboards*, que podem representar os diferentes contextos em que as redes de sensores podem estar instaladas. Em resultado do trabalho desenvolvido averigua-se que a *eneidaws* fica com um sistema modelar e flexível capaz de acolher novas funcionalidades.

Em suma, o desenvolvimento deste projeto auxiliou na análise e aprendizagem de novas tecnologias e conceitos de outras áreas, que virão a demonstrar-se importantes na formação pessoal e profissional do aluno. É, portanto, com satisfação e sentimento de trabalho concluído que se considera terminado este projeto, dado terem sido satisfeitas as expectativas da empresa envolvida e dos restantes intervenientes no projeto.

REFERÊNCIAS BIBLIOGRÁFICAS

Agile Modeling (2014). *Feature Driven Development Web site*. Acedido em Fevereiro 1 de 2014, em <http://agilemodeling.com/essays/fdd.htm>

Costa, António Casimiro (1995). *Uma Panorâmica Sobre Sistemas SCADA*. Introdução à Investigação, Mestrado em Engenharia Eletrotécnica e de Computadores Área de Computadores, Instituto Superior Técnico.

Costa, Diogo Silva (2009). *Sistema de Supervisão e Controlo Integrados*. Mestrado em Engenharia Mecânica, Universidade de Aveiro Departamento de Engenharia Mecânica

Exata (2010). *Exata, sistemas de automação. MoviconX Técnico Web site*. Acedido em Fevereiro 15 de 2014, em <http://www.moviconx.com.br/>

Feng Hua (2014). *Sistema de gestão Web site*. Acedido em Março 22 de 2014, em <http://www.gzhttp.com/IT/web/webSolution/2014/0311/68.html>

GECCAI (2014). *SCADA - Los sistemas SCADA (Supervisory Control And Data Acquisition) son aplicaciones de software Web site*. Acedido em Março 28 de 2014, em <http://gecca.wordpress.com/scada/>

Microsoft (2006). *Multi-Tenant Data Architecture Web site*. Acedido em Março 23 de 2014, em <http://msdn.microsoft.com/en-us/library/aa479086.aspx>

Microsoft (2013). *Visão geral de aplicativos de dados de N camadas Web Site*. Acedido em Março 25 de 2014, em <http://msdn.microsoft.com/pt-BR/library/bb384398.aspx>

Microsoft (2014). *SQL Server Editions Web site*. Acedido em Fevereiro 10 de 2014, em <http://www.microsoft.com/en-us/server-cloud/products/sql-server-editions/sql-server-express.aspx>

Nunes, Mauro e O'Neill, Henrique (2004). *UML, abordagem simples e prática tudo o que precisa para compreender e construir os principais diagramas* (5ª edição). FCA – Editora de Informática, Lda.

Protocolo TI (2012). *MVC - Model-View-Controller Web site*. Acedido em Março 21 de 2014, em <http://protocoloti.blogspot.pt/2012/12/mvc-model-view-controller.html>

OPC Foundation (2014). *What is OPC Web site*. Acedido em Fevereiro 15 de 2014, em <https://opcfoundation.org/about/what-is-opc/>

Quintas, António Rocha (2004). *SCADA. Automação Industrial Ramo de Automação, Produção e Eletrónica Industrial*, Faculdade de Engenharia da Universidade do Porto.

ScadaBR (2012). *ScadaBR, Sistema de Supervisão Web site*. Acedido em Fevereiro 15 de 2014, em <http://www.scadabr.org.br/?q=node/1>

Siemens AG. (2014). *Simatic Wincc. Industry automation and drive technologies – Siemens Web site*. Acedido em Fevereiro 15 de 2014, em <http://www.automation.siemens.com>

TDC2013 (2013). *Ecossistema com serviços via Web Api Web site*. Acedido em Abril 10 de 2014, em <http://www.infoq.com/br/presentations/ecossistema-web-api>

TechTerms (2014). *Thread Web site*. Acedido em Abril 11 de 2014, em <http://www.techterms.com/definition/thread>

Webopedia (2014). *SCADA - Supervisory Control and Data Acquisition Web site*. Acedido em Fevereiro 3 de 2014, em <http://www.webopedia.com/TERM/S/SCADA.html>

Wikipédia (2011). *SCRUM Web site*. Acedido em Fevereiro 1 de 2014, em [http://en.wikipedia.org/wiki/Scrum_\(software_development\)](http://en.wikipedia.org/wiki/Scrum_(software_development))

Wikipédia (2012). *SCADA Web site*. Acedido em Fevereiro 5 de 2014, em <http://en.wikipedia.org/wiki/SCADA>

Wikipédia (2013). *Posto de Transformação Web site*. Acedido em Junho 1 de 2014, em http://pt.wikipedia.org/wiki/Posto_de_transforma

Wikipédia (2014a). *ASP.NET MVC Framework Web site*. Acedido em Março 26 de 2014, em http://en.wikipedia.org/wiki/ASP.NET_MVC_Framework

Wikipédia (2014b). *ISM band Web site*. Acedido em Julho 16 de 2014, em http://en.wikipedia.org/wiki/ISM_band

ZigBee Alliance (2014). *ZigBee Specifications Web site*. Acedido em Julho 16 de 2014, em <http://www.zigbee.org/Specifications.aspx>

ANEXOS

ÍNDICE DE ANEXOS

Anexo A - Análise de Requisitos	A-1
Anexo B - Prototipagem das Aplicações	A-9
Anexo C - Modelo de Dados	A-21
Anexo D - Manual de utilizador	A-27
Anexo E - Testes	A-65

ANEXO A

Análise de Requisitos

Diagramas de Caso de Uso

1. DIAGRAMAS DE CASO DE USO

Nesta secção apresentam-se os diagramas de casos de uso adicionais com informação complementar referente ao diagrama de pacotes da figura A-1.

1.1. DIAGRAMAS DE CASOS DE USO APLICAÇÃO SCADA WEB

No diagrama de pacotes e casos de uso da figura A-1, apresenta-se de forma esquemática os requisitos gerais das funcionalidades da aplicação *SCADA Web* em pacotes e os respetivos atores que intervêm nesta aplicação.

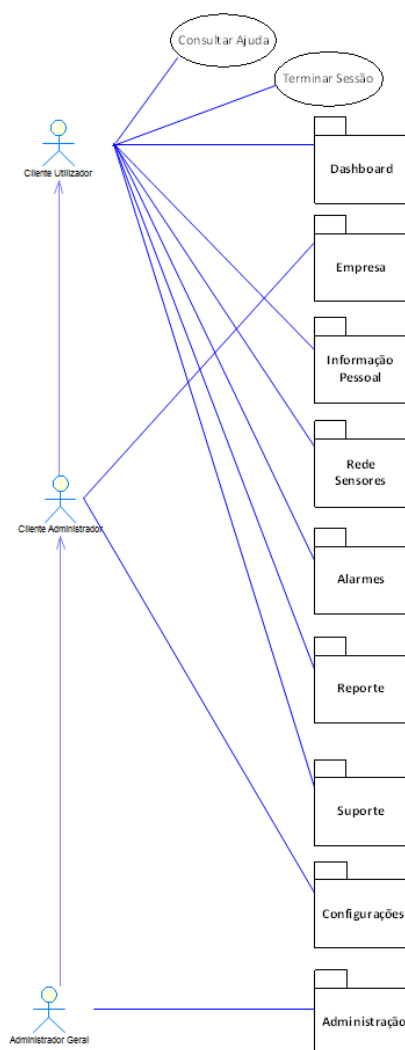


Figura A-1 - Diagrama de Pacotes e Casos de Uso das Funcionalidades Gerais Aplicação SCADA Web

No diagrama de casos de uso referente ao pacote *dashboard*, visto na figura A-2, apresenta-se de forma esquemática os casos de uso e os atores intervenientes. O pacote *links* representa os diferentes pacotes de opções gerais vistas na figura A-1, a que cada ator pode aceder mediante os respetivos níveis de privilégios. Os atores podem ainda visualizar o número de alarmes críticos, número de alarmes alerta e o número de mensagens recebidas.

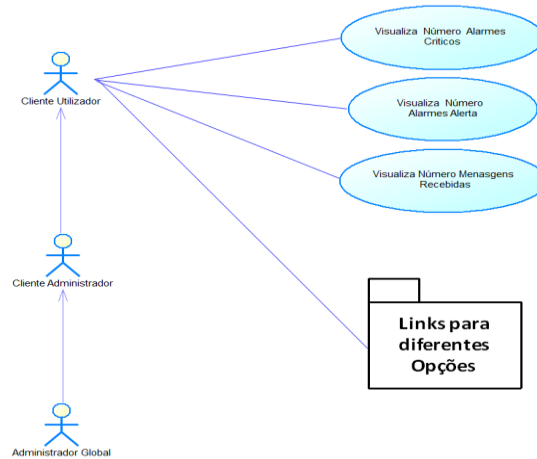


Figura A-2 - Casos de uso do pacote Dashboard

No diagrama de casos de uso referente ao pacote *Empresa*, visto na figura A-3, apresenta-se de forma esquemática os casos de uso e os atores intervenientes. O ator cliente administrador pode consultar os dados da sua empresa e atualizar os dados da mesma.



Figura A-3 - Casos de uso do pacote Empresa

No diagrama de casos de uso referente ao pacote *Informações Pessoais*, visto na figura A-4, apresenta-se de forma esquemática os casos de uso e os atores intervenientes. Os atores podem consultar as suas informações pessoais e opcionalmente podem atualizar os seus dados ou alterar as suas *password*.

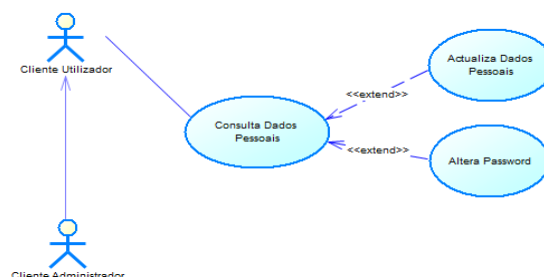


Figura A-4 - Casos de uso do pacote Informações Pessoais

No diagrama de casos de uso referente ao pacote *Rede de Sensores*, visto na figura A-5, apresenta-se de forma esquemática os casos de uso e os atores intervenientes. Os atores podem visualizar as redes de sensores através de um mapa ou tabela. Opcionalmente podem pesquisar por uma rede e visualizar todas as redes com alarmes críticos, alarmes alerta e todas as redes sem alarme. Podem ainda escolher uma rede e visualizar os últimos dados adquiridos numa rede.

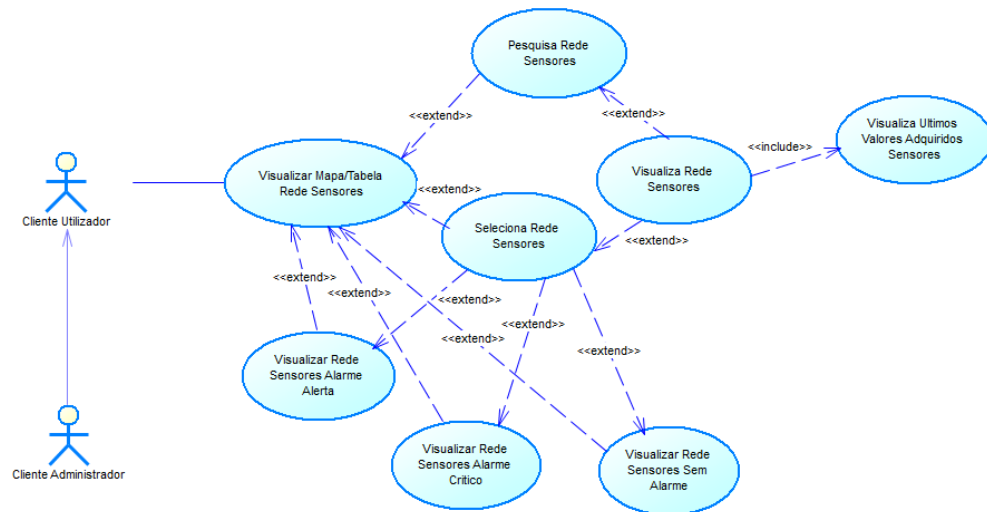


Figura A-5 - Caso de uso Rede de Sensores

No diagrama de casos de uso referente ao pacote *Alarmes*, visto na figura A-6, demonstra-se de forma esquemática os casos de uso e os atores intervenientes. Os atores podem consultar a lista de alarmes e visualizar os dois géneros de alarmes críticos e alertas. Para validação de um alarme só o ator cliente administrador pode definir um alarme como tratado.

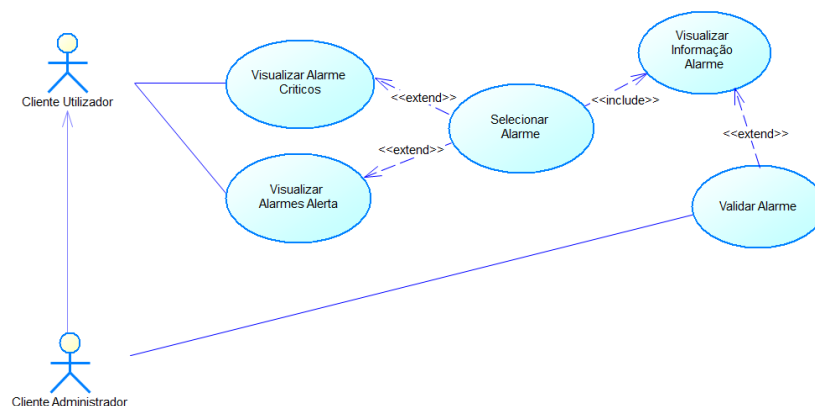


Figura A-6 - Casos de uso do pacote Alarme

No diagrama de casos de uso referente ao pacote *Reporte*, visto na figura A-7, demonstra-se de forma esquemática os casos de uso e os atores intervenientes. Todos os atores podem listar as redes se sensores e escolher um sensor para consulta do histórico de dados em formato gráfico ou tabela. Opcionalmente os atores podem fazer pesquisas customizadas sobre os dados e informações das redes de sensores.

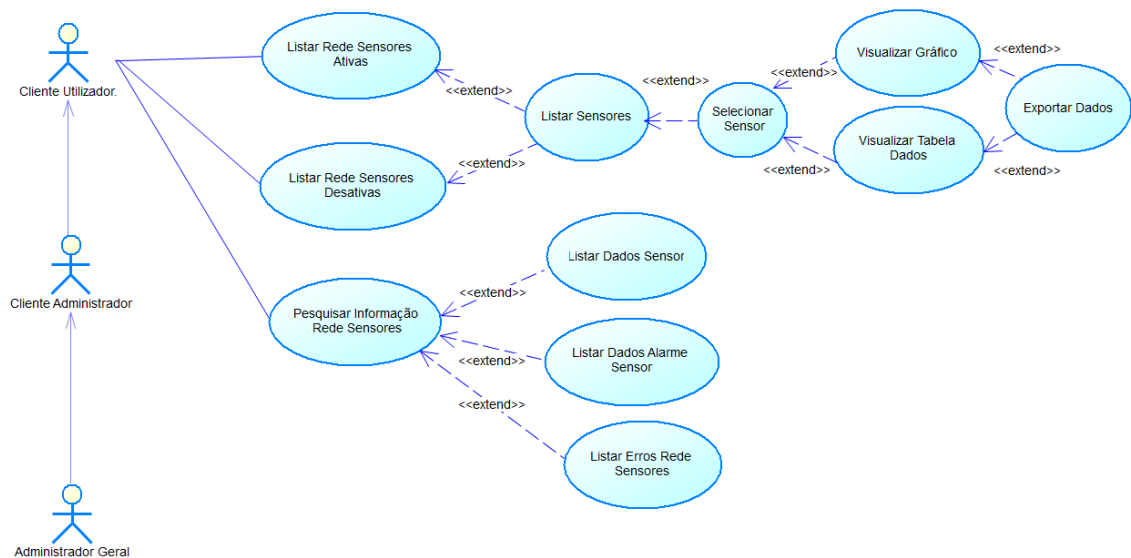


Figura A-7 - Casos de uso do pacote *Reporte*

No diagrama de casos de uso referente ao pacote *Suporte*, visto na figura A-8, demonstra-se de forma esquemática os casos de uso e os atores intervenientes. Os atores podem opcionalmente criar mensagens para pedir ajuda sobre a aplicação *Web* ou listar todas as mensagens escolhendo uma mensagem para ler ou para apagar.

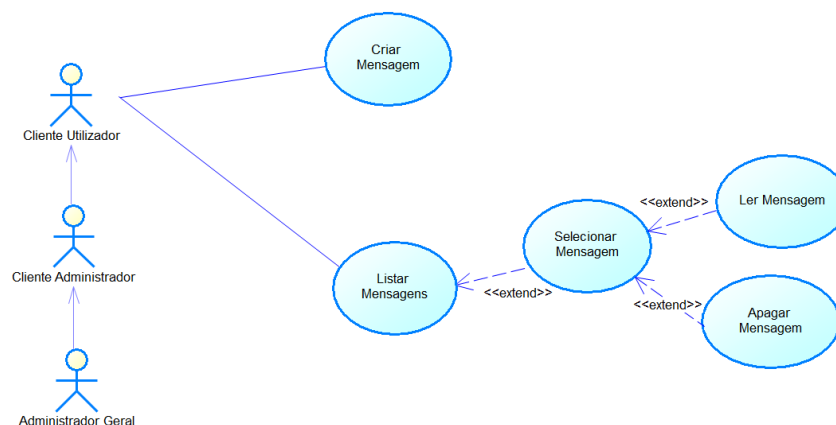


Figura A-8 - Casos de uso do pacote *Suporte*

No diagrama de casos de uso referente ao pacote *Configurações*, visto na figura 9, demonstra-se de forma esquemática os casos de uso e os atores intervenientes. Os atores podem escolher as seguintes opções: criar redes de sensores, criar grupos, criar modelos, criar regras para alarmes. Para todas estas opções o ator pode listar, seleccionar, editar, desativar e apagar.

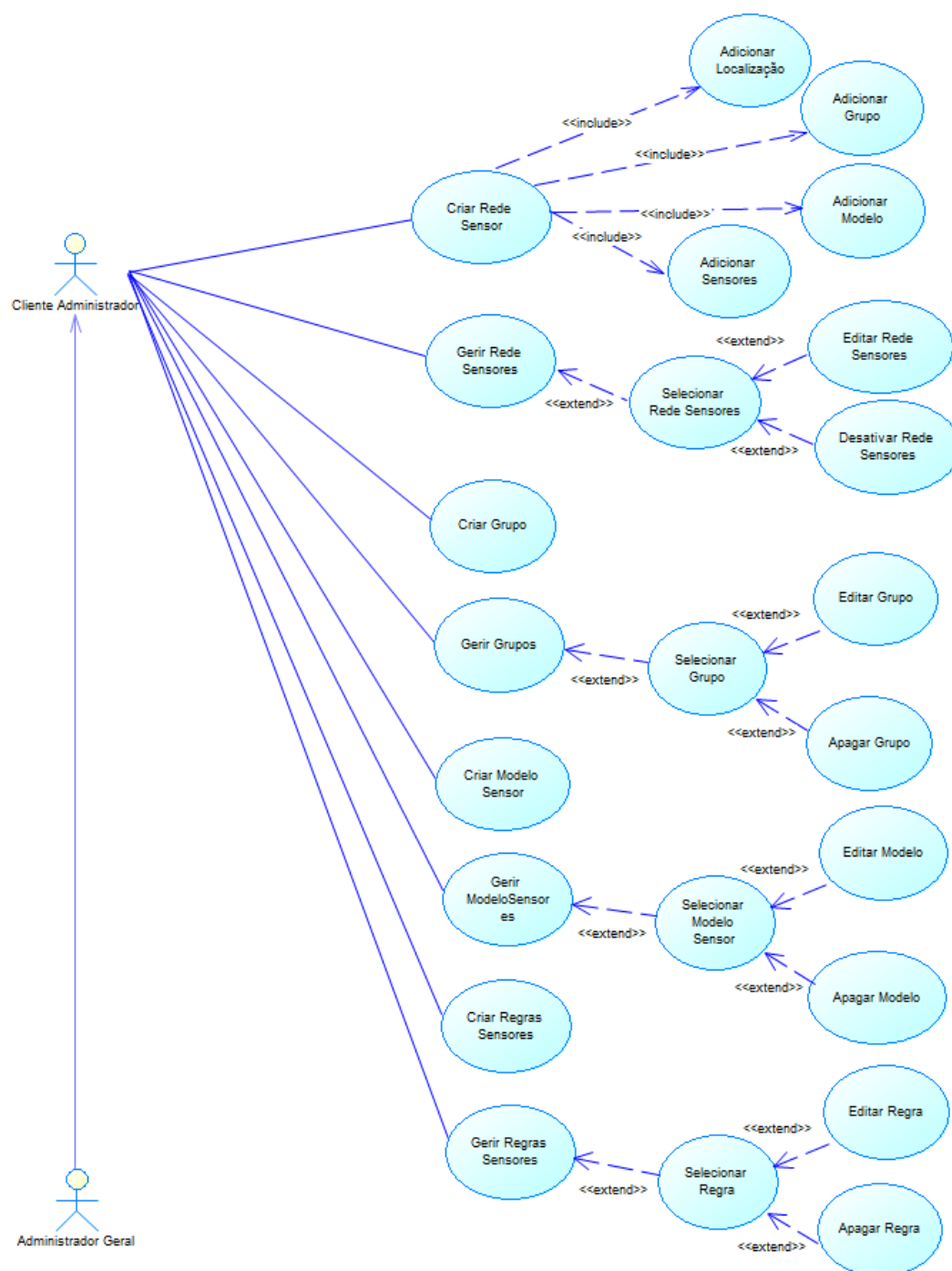


Figura A-9 - Casos de uso do pacote *Configurações*

No diagrama de casos de uso referente ao pacote *Administração*, visto na figura 10, demonstra-se de forma esquemática os casos de uso e os atores intervenientes. O ator administrador geral, para criar uma empresa na aplicação *Web*, precisa criar a empresa com os seus dados e obrigatoriamente criar uma licença e um ator cliente administrador responsável por gerir o acesso à aplicação *Web*. Este ator pode criar empresas, licenças e utilizadores; para cada um destes, o ator pode listar, editar e apagar informações.

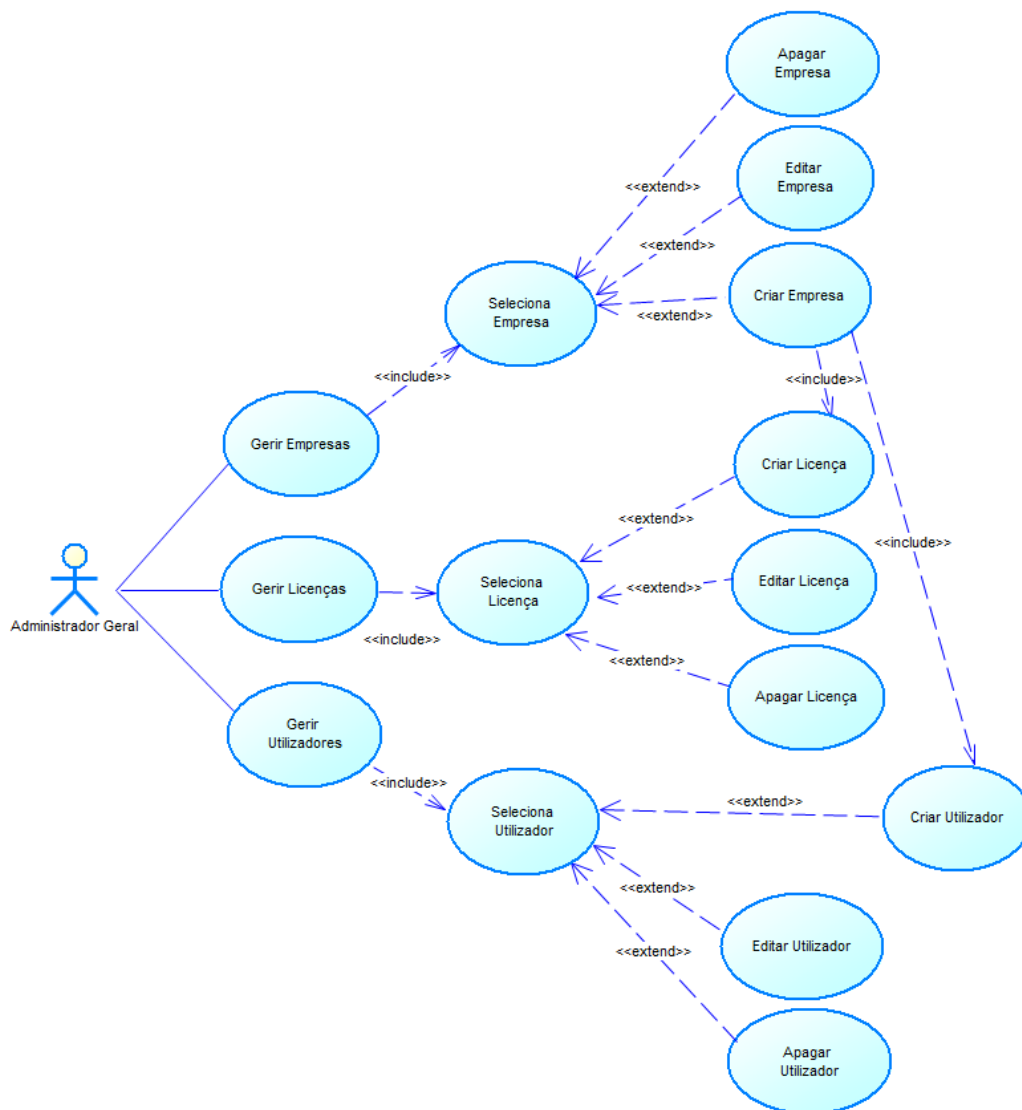


Figura A-10 - Casos de uso do pacote Administração

ANEXO B

Protótipos

Protótipos das Aplicações

1. PROTÓTIPO DAS APLICAÇÕES

Nesta seção apresentam-se os *mockups* dos protótipos definidos para cada aplicação. Estes protótipos serviram para validação da proposta feita pelo aluno à empresa e para perceber qual o aspeto que as aplicações iriam apresentar e o caminho a seguir no desenvolvimento gráfico com as respetivas funcionalidades e opções.

1.1. PROTÓTIPO APLICAÇÃO SERVIDOR

Na figura B-1 apresenta-se o *mockup* da aplicação servidor. Nesta figura pode ver-se as opções do menu: *Ficheiro*, *Opções* e *Ajuda*. A opção *Ficheiro* permite ao utilizador escolher a opção para sair da aplicação. A opção *Opções* permite ao utilizador configurar o *IP*, porto da aplicação servidor e ativar ou desativar a verificação de alarmes. A opção *Ajuda* permite ao utilizador visualizar o manual de ajuda da aplicação servidor.

No quadrado preto do lado esquerdo da figura B-1, pode ver-se o *IP* e porto da aplicação servidor e a respetiva lista de redes de sensores ligadas. O quadrado preto do lado direito permite ver o *log* de todas as mensagens trocadas com o servidor. Os três botões do lado direito permitem minimizar, maximizar e fechar a aplicação, e os botões do lado esquerdo permitem iniciar, parar e limpar *log* de mensagens da aplicação.

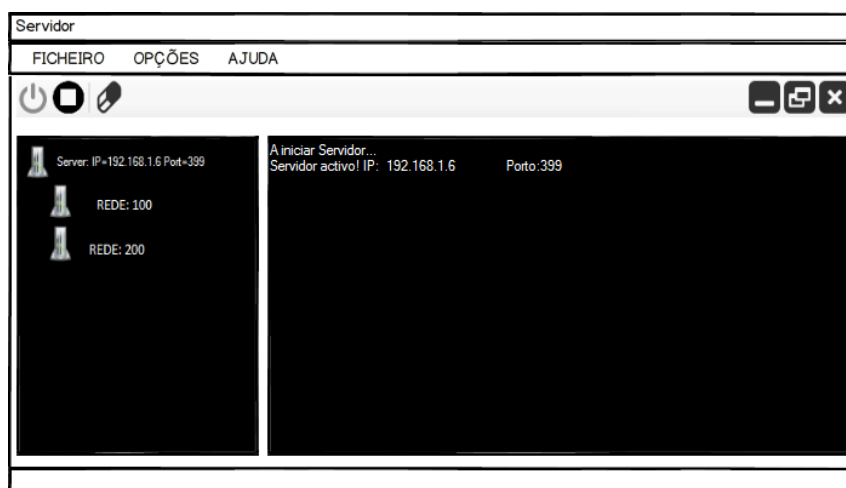


Figura B-1 - Protótipo Gráfico da Aplicação Servidor

1.2. PROTÓTIPO APLICAÇÃO SERVIÇO DE ALARMES

Na figura B-2 apresenta-se o *mockup* da aplicação serviço de alarmes. Nesta figura pode ver-se as opções menu: *Ficheiro* e *Ajuda*. A opção *Ficheiro* permite ao utilizador escolher a opção para sair da aplicação. A opção *Ajuda* permite o utilizador visualizar o manual de ajuda da aplicação servidor. O utilizador, para colocar o serviço de alarmes em funcionamento, tem de definir um relógio, para ser feita a verificação de alarmes com a periodicidade que este definir. Por exemplo, se um relógio for definido com 30 segundos, isso significa que de 30 em 30 segundos a aplicação estará à procura de alarmes. Os três botões do lado esquerdo permitem iniciar, parar e limpar *log* da aplicação. Se a aplicação estiver em funcionamento, aparece a mensagem “Sistema Ligado”; em caso contrário, aparece “Sistema Parado”.

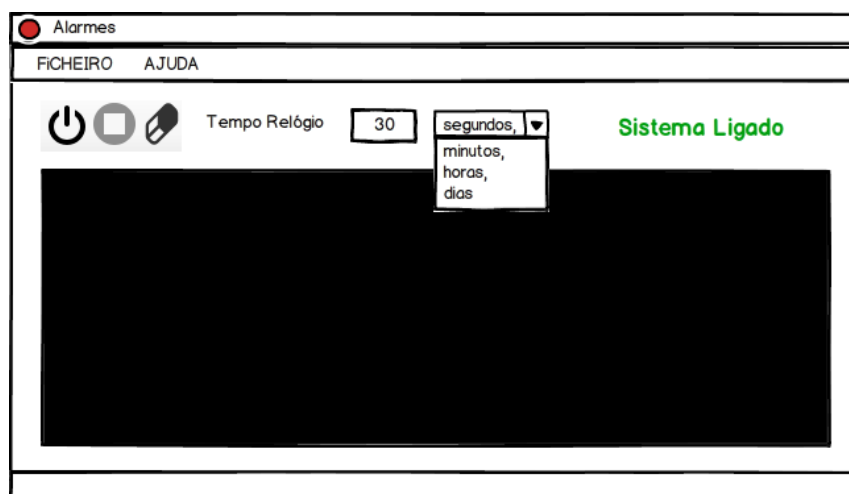


Figura B-2 – Protótipo Gráfico da Aplicação Serviço de Alarmes

1.3. PROTÓTIPO APLICAÇÃO SCADA WEB

As figuras que se seguem representam todos os *mockups* definidos para a aplicação *SCADA Web*. Na figura B-3 apresenta-se o *mockup* da página inicial da aplicação *Web*, onde o utilizador pode fazer as seguintes ações: *login* na aplicação para acesso ao sistema *SCADA*; recuperação da *password* por *e-mail*, na eventualidade de este ter esquecido a mesma; e, por fim, consulta da página com todos contactos da *eneidaws*.

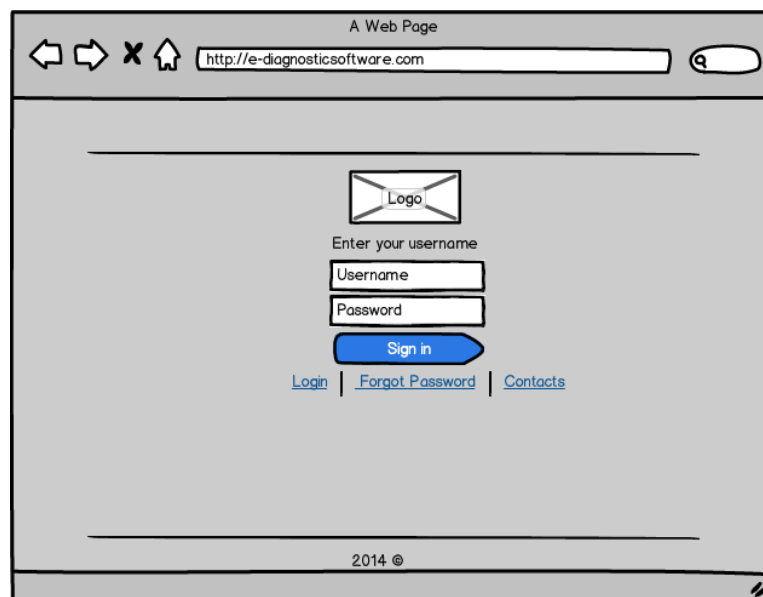


Figura B-3 - Aplicação Web Página de Login

Na figura B-4 apresenta-se o *mockup* da página *Dashboard* da aplicação *Web*. Esta página contém *links* para acesso às diferentes opções vistas do lado esquerdo da figura. A representação de cada *link* é feita através do ícone que se adequa mais à respetiva opção. Esta página tem ainda informações rápidas, como: número de alarmes críticos existentes nas redes de sensores; número de alarmes alerta existentes nas redes de sensores; número de redes de sensores configuradas e número de mensagens de suporte que existem por ler. O conteúdo desta página varia de acordo com os níveis de privilégios do utilizador, ficando visíveis mais ou menos opções.

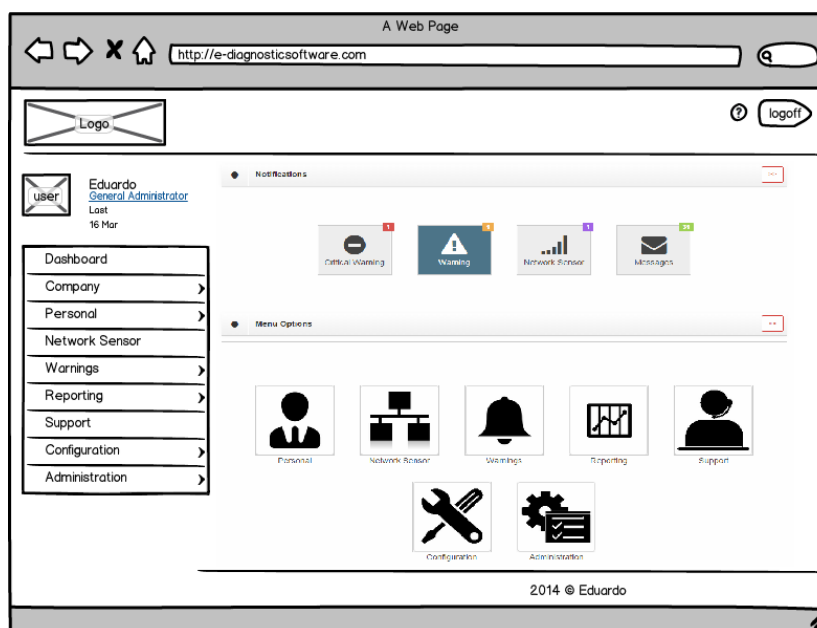


Figura B-4 - Aplicação Web Página Dashboard

Na figura B-5 apresenta-se o *mockup* da página *Empresa* da aplicação *Web*. Esta página contém todos os dados da empresa configurada, sendo possível fazer *upload* do logótipo da empresa.

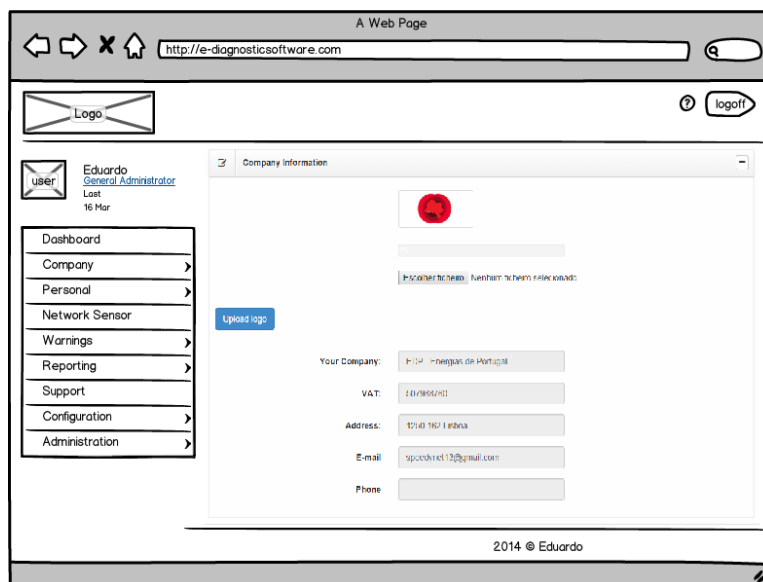


Figura B-5 - Aplicação Web Página com Informação da Empresa

Na figura B-6 apresenta-se o *mockup* da página *Informação Pessoal do Utilizador* da aplicação *Web*. Esta página contém todos os dados do utilizador, sendo possível fazer *upload* da fotografia do utilizador ou atualizar algumas das suas informações.

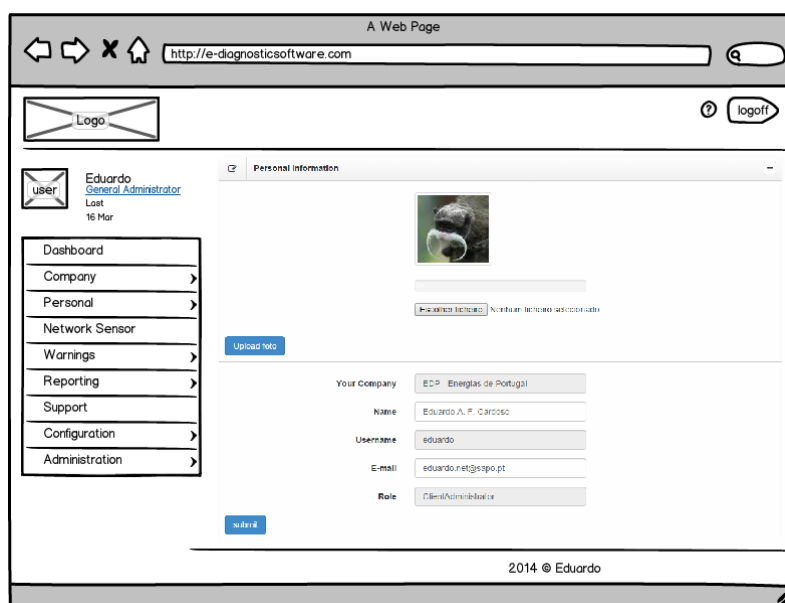


Figura B-6 - Aplicação Web Página com Informação Pessoal do Utilizador

Na figura B-7 apresenta-se o *mockup* da página *Informação Pessoal do Utilizador (alterar password)* da aplicação Web. Esta página permite ao utilizador modificar a sua *password* pessoal sempre que pretenda.

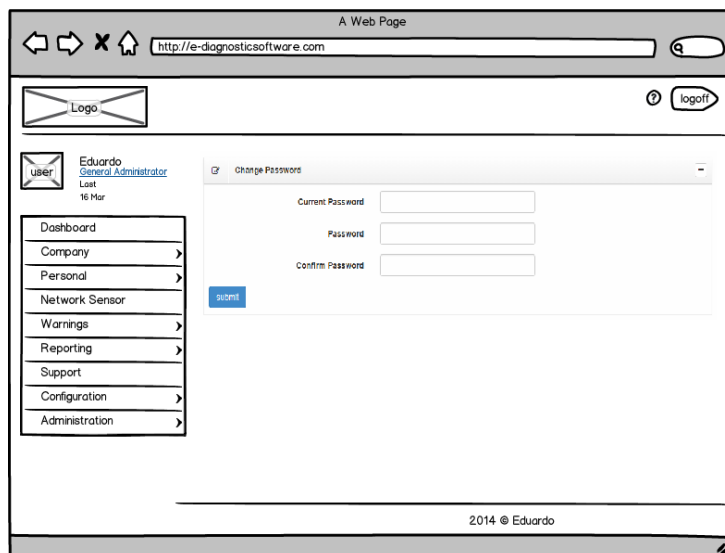


Figura B-7 - Aplicação Web Página para o utilizador alterar a sua password

Na figura B-8 apresenta-se o *mockup* da página *Rede de Sensores* da aplicação Web. Esta página permite o utilizador ver as redes de sensores configuradas por coordenadas GPS. A representação de cada rede é feita por um ponto no mapa que pode apresentar três cores (verde, vermelho e amarelo): verde significa que não existem alarmes na rede, amarelo ou vermelho significa que existe um alarme alerta ou alarme crítico na rede de sensores. O ponto da rede de sensores deve ter também um *link* para o *dashboard* da rede, onde é possível ver todos os sensores configurados nessa rede.

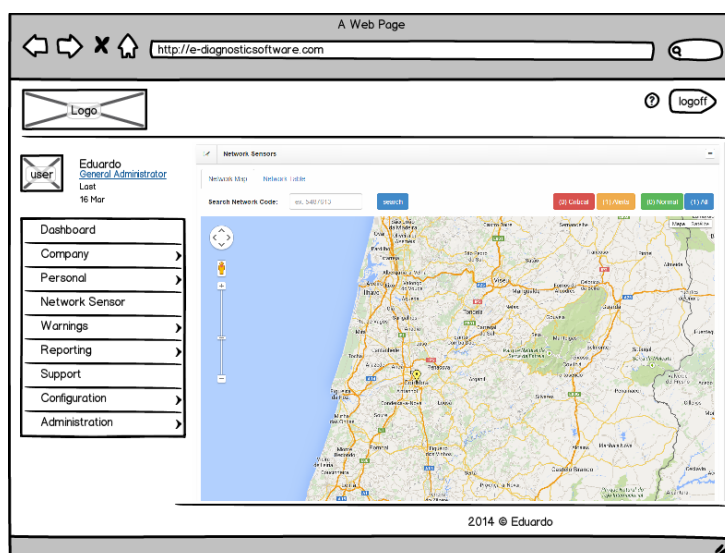


Figura B-8 - Aplicação Web Página para visualizar as redes de sensores num mapa por coordenadas GPS

Na figura B-9 apresenta-se o *mockup* da página *Dashboard da Rede de Sensores* da aplicação Web. Esta página permite ao utilizador visualizar os últimos dados adquiridos pela rede de sensores e saber o estado de cada um dos sensores (se estes se encontram em alarme ou não).

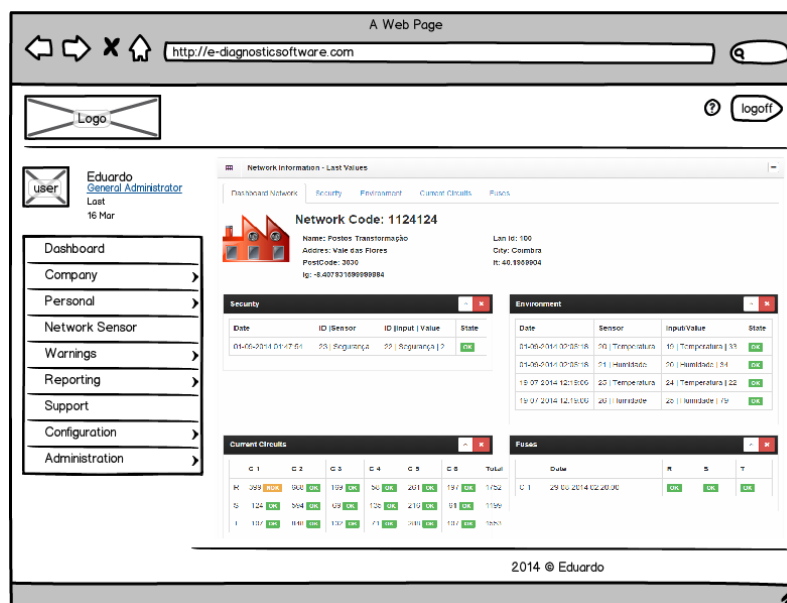


Figura B-9 - Aplicação Web Página para utilizador visualizar os últimos dados numa rede de sensores

Na figura B-10 apresenta-se o *mockup* da página *Alarmes* da aplicação Web. Esta página permite ao utilizador consultar todos os alarmes ocorridos nas redes de sensores, e ainda ver em detalhe a razão que gerou determinado alarme.

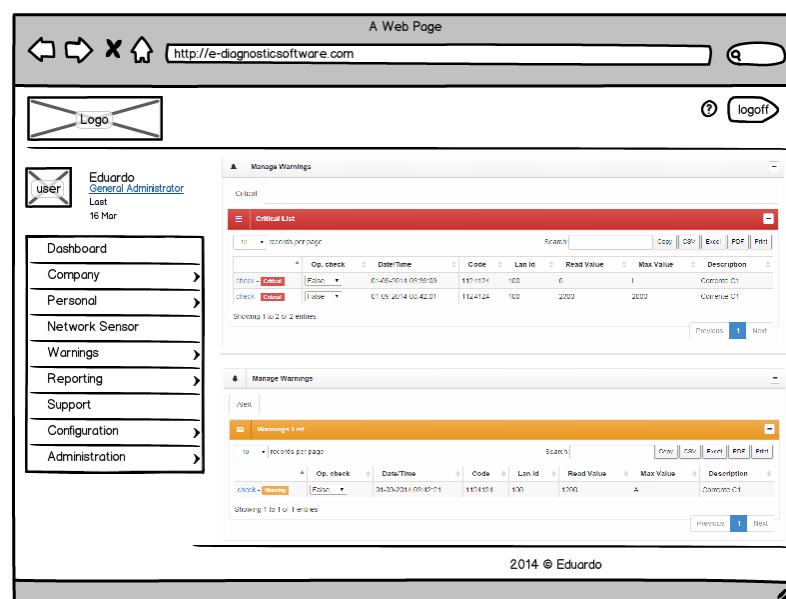


Figura B-10 - Aplicação Web Página para visualizar os alarmes ocorridos nas diferentes redes

Na figura B-11 apresenta-se o *mockup* da página *Reporte* da aplicação Web. Esta página permite ao utilizador consultar o histórico com a informação de todos os dados adquiridos pelos vários sensores de uma rede. O utilizador pode consultar o histórico de dados através de gráficos (figura B-12) ou tabelas. A página *Reporte* só está disponível para clientes da *eneidaws* que tenham licença válida para acesso a esta opção.

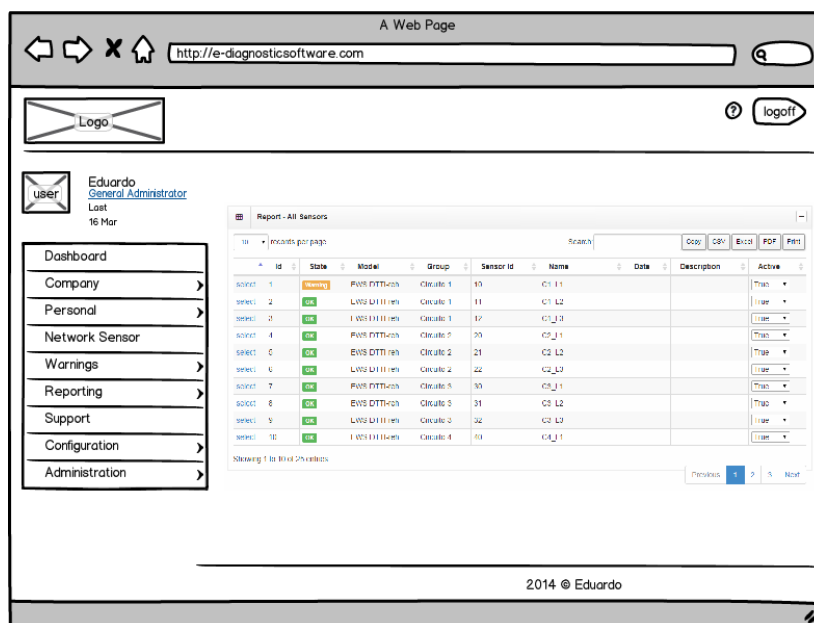


Figura B-11 - Aplicação Web Página para escolha da rede de sensores e visualização do histórico de dados



Figura B-12 - Aplicação Web Página para representação dos dados dos sensores através de gráficos

Na figura B-13 apresenta-se o *mockup* da página *Suporte* da aplicação *Web*. Esta página permite o utilizador consultar as suas novas mensagens colocadas pelos utilizadores que necessitam de suporte e criar mensagens de ajuda.

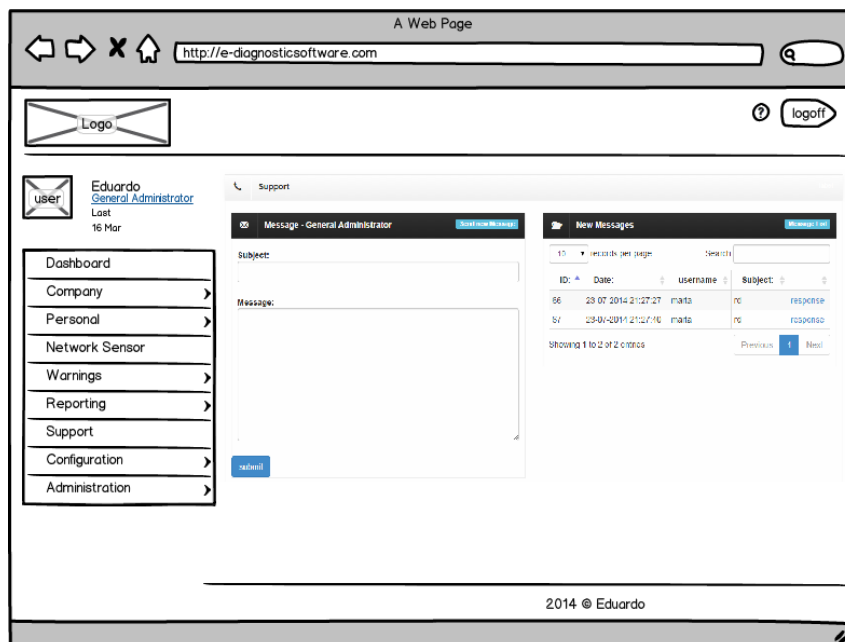


Figura B-13 - Aplicação Web Página de suporte para os utilizadores contactarem com o administrador

Na figura B-14 apresenta-se o *mockup* da página *Configuração* da aplicação *Web*. Esta página permite ao utilizador criar uma rede de sensores com base num *wizard*. Com este *wizard* o utilizador cria a localização da rede de sensores, os dados da rede de sensores e a configuração dos respetivos sensores.

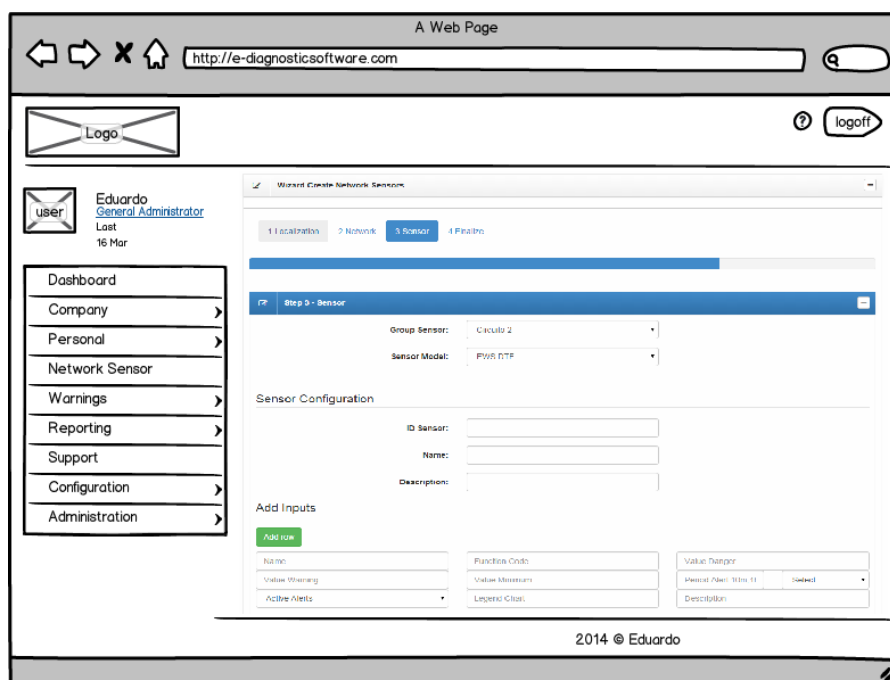


Figura B-14 - Aplicação Web Página com wizard para configuração das redes de sensores

Na figura B-15 apresenta-se o *mockup* da página *Administração (Criar Empresa)*, da aplicação *Web*. Esta página permite ao utilizador administrador da *eneidaws* criar empresas que terão acesso à aplicação *Web*. Nesta página são definidos todos os dados da empresa.

The image shows a web browser window titled "A Web Page" with the address bar displaying "http://e-diagnosticsoftware.com". The page layout includes a top navigation bar with a "Logo" placeholder and a "logout" button. On the left, a sidebar menu lists various administrative functions: Dashboard, Company, Personal, Network Sensor, Warnings, Reporting, Support, Configuration, and Administration. The main content area is titled "Create Company" and contains a form with the following fields: Country (a dropdown menu), Name, VAT, Email, Phone, Address, Database, Ip, Dbusername, and Dbpassword. A "submit" button is located at the bottom left of the form. The footer of the page displays "2014 © Eduardo".

Figura B-15 - Aplicação Web Página para criar empresas

Na figura B-16 apresenta-se o *mockup* da página *Administração (Criar Licença)*, da aplicação *Web*. Esta página permite ao utilizador administrador da *eneidaws* criar licenças para as empresas que terão acesso à aplicação *Web*. Esta licença diferencia as empresas que têm acesso a algumas opções da aplicação *Web* ou a todas.

The image shows a web browser window titled "A Web Page" with the address bar displaying "http://e-diagnosticsoftware.com". The page layout is consistent with the previous figure, featuring a top navigation bar with a "Logo" placeholder and a "logout" button, and a sidebar menu with the same administrative functions. The main content area is titled "Create Licence" and contains a form with the following fields: Company (a dropdown menu), Type License, Initial Date (pre-filled with "01-01-2008"), End Date (pre-filled with "01-01-2008"), Version, and Description. A "submit" button is located at the bottom left of the form. The footer of the page displays "2014 © Eduardo".

Figura B-16 - Aplicação Web Página para criar licenças

Na figura B-17 apresenta-se o *mockup* da página *Administração (Criar utilizador)* da aplicação *Web*. Esta página permite ao utilizador administrador da *eneidaws* criar o utilizador Administrador da empresa do seu cliente que terá acesso à aplicação *Web*. Este utilizador será responsável por gerir toda a sua informação na aplicação *Web*, referente à sua empresa.

Em suma, todo o processo necessário que o administrador seguir para criar uma empresa na aplicação *Web* passa pelos seguintes passos: criar empresa, criar licença para a empresa e criar o utilizador administrador dessa empresa.

The mockup shows a web browser window with the title "A Web Page" and the address bar displaying "http://e-diagnosticsoftware.com". The page layout includes a top navigation bar with a "Logo" placeholder and a "logoff" button. A sidebar on the left contains a user profile for "Eduardo General Administrator" (last active 16 Mar) and a menu with the following items: Dashboard, Company, Personal, Network Sensor, Warnings, Reporting, Support, Configuration, and Administration. The main content area is titled "Create User" and features a form with the following fields: Name, Username, E-mail, Role (a dropdown menu with "Choose Role" selected), and Company (a dropdown menu with "Choose Company" selected). A blue "submit" button is positioned below the form fields. The footer of the page displays "2014 © Eduardo".

Figura B-17 - Aplicação Web Página para criar utilizadores

ANEXO C

Modelo de Dados

Modelo Conceptual e Modelo Físico

1. DIAGRAMA BASE DE DADOS GERAL

Nesta secção apresenta-se o diagrama entidade relacionamento (figura C-1) e o modelo físico (figura C-2) da base de dados geral.

1.1. DIAGRAMA ENTIDADE RELACIONAMENTO

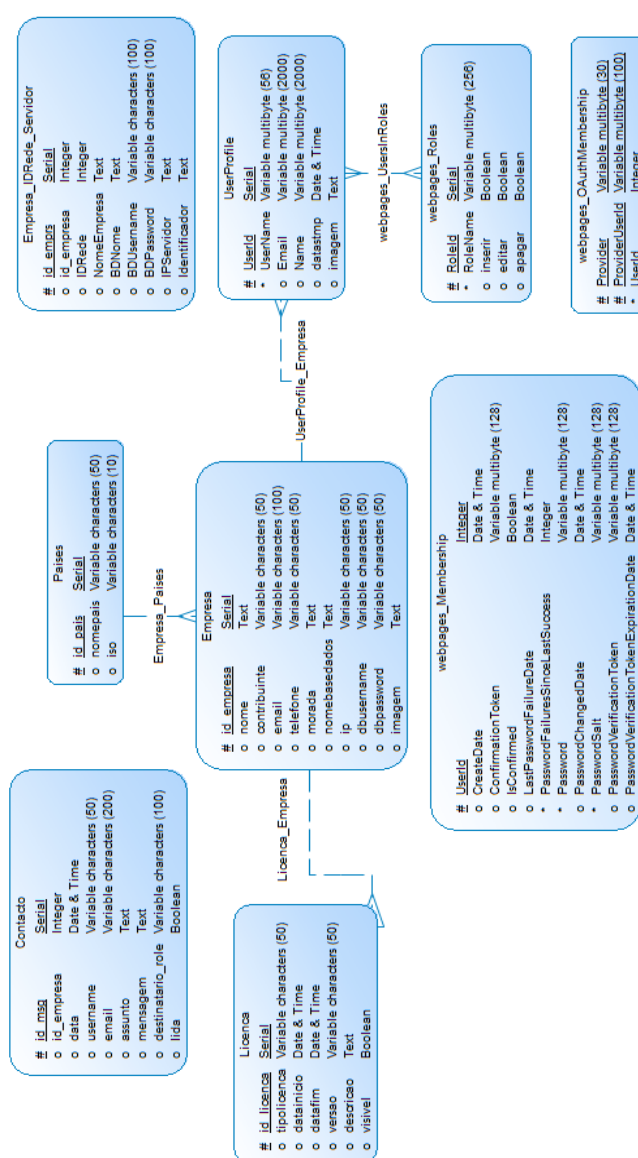


Figura C-1 - Modelo Entidade Relacionamento Base de Dados Geral

1.2. MODELO FÍSICO

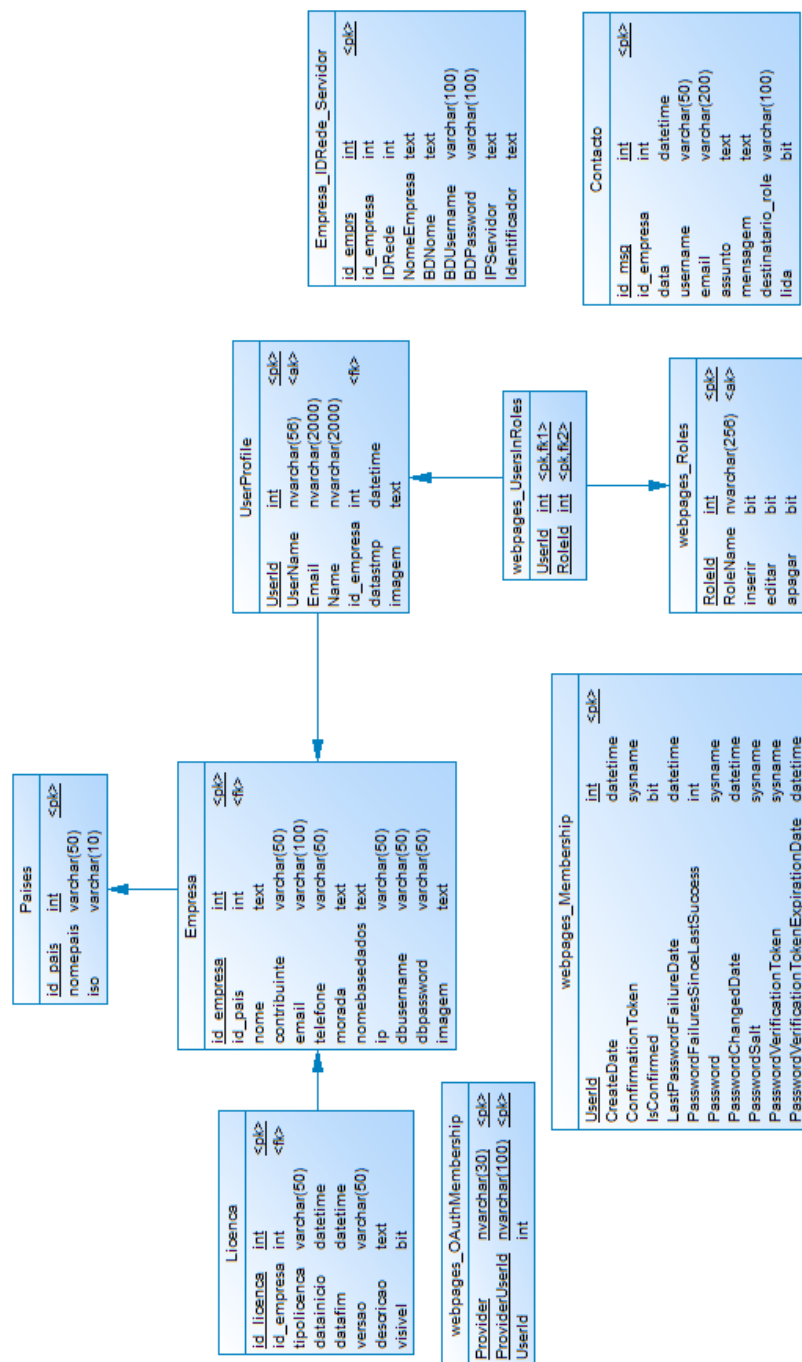


Figura C-2 - Modelo Físico Base de Dados Geral

ANEXO D

Manual de Utilizador das Aplicações

1. MANUAL UTILIZADOR APLICAÇÃO SERVIDOR

Nesta secção apresenta-se o manual de utilizador da aplicação servidor. Esta aplicação é configurada pelo administrador da *eneidaws*, sendo este utilizador responsável por gerir todo o sistema *e-Diagnostic Software* para monitorização de sensores. Esta é uma aplicação *Windows Forms* executável com funcionamento local em sistemas operativos *Windows*. Esta aplicação, antes de ser usada, tem um ficheiro *XML* onde é parametrizada a *string* de conexão à base de dados, para a aplicação servidor guardar os dados das redes de sensores.

Na figura D-1, podem ver-se as opções da aplicação servidor após esta aplicação ser executada. A opção *Ficheiro* permite ao utilizador escolher a opção “sair da aplicação”. A opção *Ajuda* permite ao utilizador consultar informações sobre ajuda da aplicação e créditos da aplicação. A opção *Opções* contém as várias funcionalidades da aplicação servidor. Do lado esquerdo da figura D-1, podem ver-se as redes clientes ligadas ao servidor, e do lado direito podemos ver as mensagens no *log* trocadas com o servidor. Os ícones dos botões do lado esquerdo permitem ligar ou desligar o servidor e limpar *log* das mensagens. O servidor só pode ser ligado para escuta de mensagens quando é definido um *IP* e um porto, caso contrário é enviada uma mensagem de erro ao utilizador.

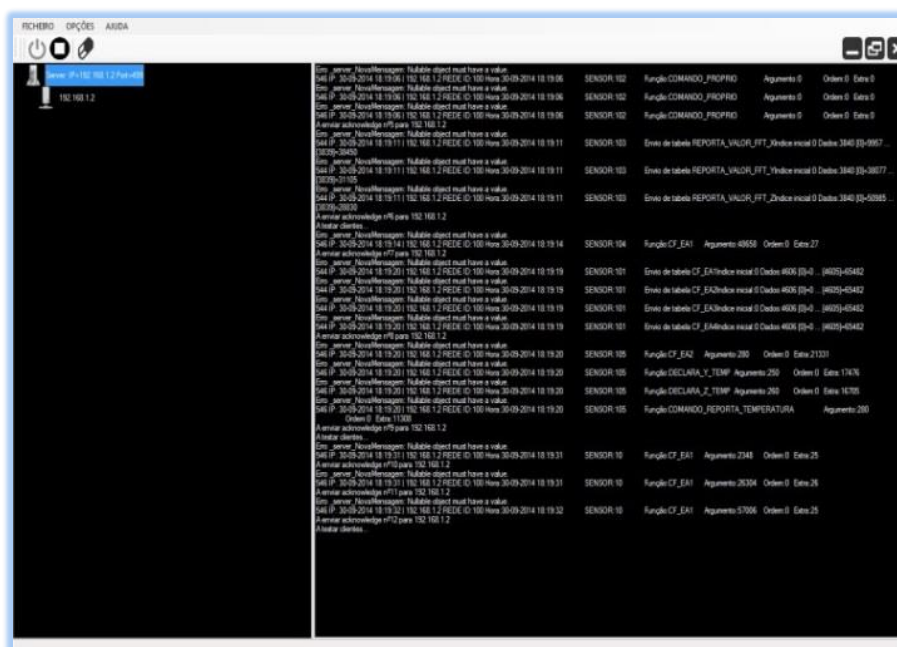


Figura D-1 - Visão Geral Aplicação Servidor

Na figura D-2 veem-se as opções do servidor. A opção *Configurações* é onde se define o *IP*, porto do servidor e, opcionalmente, onde é ativada ou desativada a verificação de alarmes, como se vê na figura D-3. A opção *Mensagens* permite ao utilizador enviar mensagens de teste para as redes de sensores, para saber quais os sensores existentes e ativos numa rede. A opção *Clientes Ligados* permite listar e ver no *log* os clientes ligados ao servidor. A opção *Log Servidor* permite ligar ou desligar *log* mensagens mediante a escolha do utilizador. Quando esta aplicação é minimizada fica a funcionar em *background*, ficando exclusivamente visível na barra de tarefas, como se pode ver na figura D-5; para maximizar, bastará clicar no ícone da aplicação.

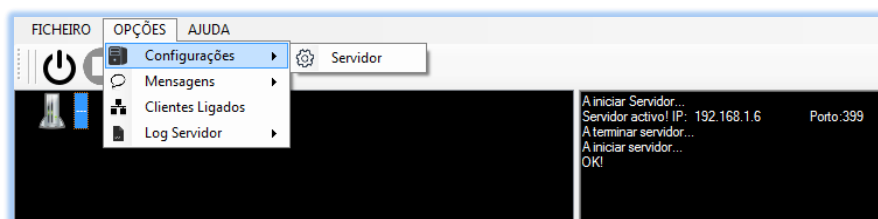


Figura D-2 - Opções Aplicação Servidor

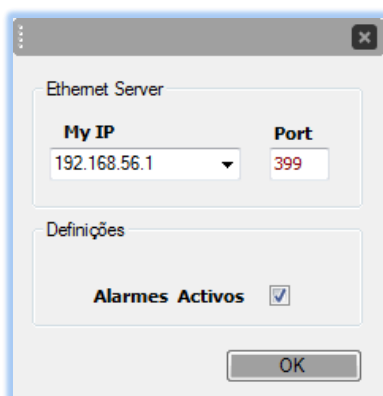


Figura D-3 - Configuração IP, Porto e ativar ou desativar visto para Alarmes

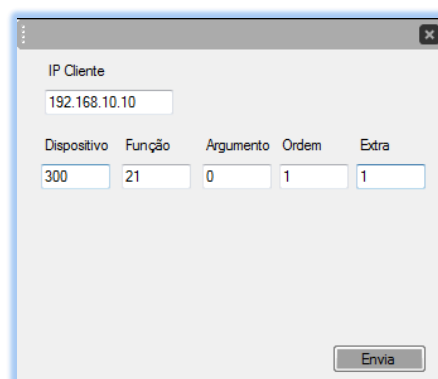


Figura D-4 - Janela Para envio de Mensagens de Testes



Figura D-5 - Ícone da aplicação servidor na barra de tarefas

Em suma, para usar a aplicação servidor, será necessário instalar a aplicação e definir a *string* de conexão para o servidor de base de dados. Ao executar a aplicação e antes de ligar o servidor para escuta de mensagens, será necessário configurar um *IP* e um porto para o servidor. Opcionalmente, o utilizador pode ativar ou desativar a verificação de alarmes por parte da aplicação servidor.

2. MANUAL UTILIZADOR APLICAÇÃO SERVIÇO ALARMES

Nesta secção apresenta-se o manual de utilizador da aplicação serviço de alarmes. Esta aplicação é usada e parametrizada pelo administrador da *eneidaws*, utilizador responsável por gerir todo o sistema *e-Diagnostic Software* para monitorização de sensores em unidades industriais. Esta é uma aplicação *Windows Forms* executável com funcionamento local em sistemas operativos *Windows*. Esta aplicação, antes de ser usada, tem um ficheiro *XML* no qual é parametrizada a *string* de conexão à base de dados, onde a aplicação serviço de alarmes irá verificar a existência de alarmes.

Na figura D-6, podem ver-se as opções da aplicação serviço de alarmes após esta aplicação ser executada. A opção *Ficheiro* permite ao utilizador escolher a opção sair da aplicação. A opção *Ajuda* permite ao utilizador consultar informações sobre ajuda da aplicação e créditos da aplicação. Como se pode ver na figura D-6, o sistema para verificação de alarmes está parado, sendo necessário o utilizador configurar o relógio para verificação de alarmes, escolhendo o tempo para que ocorra essa verificação (por exemplo de 60 em 60 segundos).

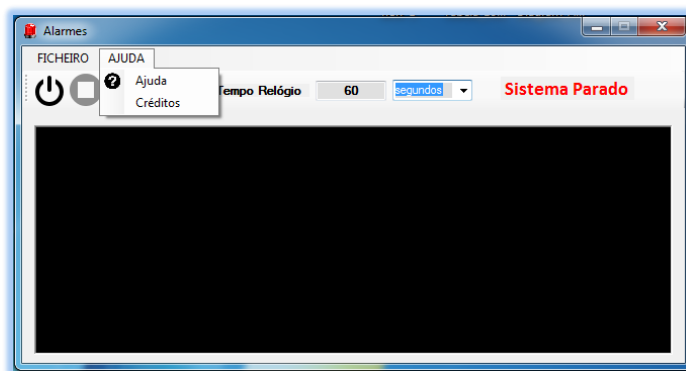


Figura D-6 - Aplicação Serviço de Alarmes Sistema Parado

Na figura D-7, o serviço já se encontra ligado porque o utilizador parametrizou o relógio com o tempo pretendido para que seja feita a verificação da existência de alarmes. Os três ícones dentro do quadrado vermelho têm o seguinte significado: ligar serviço de alarmes, parar serviço de alarmes e limpar *log*. Sempre que é enviado um *e-mail* e *SMS*, é feito um registo no *log* dessa informação no quadrado preto; sempre que o utilizador o pretenda, pode limpar essa informação. Quando esta aplicação é minimizada, fica a funcionar em *background*, ficando exclusivamente visível na barra de tarefas, como se pode ver na figura D-8; para maximizar, basta clicar no ícone da aplicação.

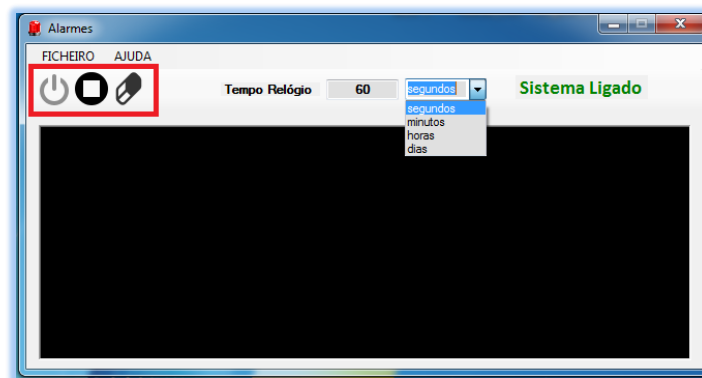


Figura D-7 - Aplicação Serviço de Alarmes Sistema Ligado



Figura D-8 - Ícone da aplicação Serviço de alarmes na barra de tarefas

Em suma, para usar a aplicação serviço de alarmes, será necessário instalar a aplicação e definir a *string* de conexão para o servidor de base de dados. Ao executar a aplicação e antes de ligar o serviço de alarmes, será necessário definir o relógio para verificação da periodicidade da verificação de alarmes (segundos, minutos, horas ou dias).

3. MANUAL WEB SERVICE

Nesta secção apresenta-se o manual da aplicação *Web Service*. Esta aplicação tem como objetivo ser usada pelos clientes da *eneidaws* que pretendam obter informação das redes de sensores por pedidos *HTTP* ao *Web Service* em formato *JSON*. Este *Web Service* é constituído por treze pedidos *GET* e por um pedido *POST*, os quais serão apresentados em seguida.

Todos os pedidos do *Web Service* funcionam com a seguinte estrutura base nos pedidos *HTTP* por *GET* ou *POST*.

Pedido público [*http://domino.net/api/webservice/*] [*Nome Método*] [*Chave*]

O pedido é constituído por domínio *DNS* para *Web Service*, nome do método a consultar e chave de acesso ao *Web Service* definido pelo administrador da *eneidaws*, para cada cliente.

GetAllLocalizacoes

Este método serve para consultar todas as localizações que as diferentes redes de sensores de uma empresa têm. Na figura D-9, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*.

<http://dominio.net/api/webservice/GetAllLocalizacoes/chave>

```
[  
  - {  
    id_local: 1,  
    nomelocal: "Coimbra",  
    descricao: "Vale das Flores"  
  }  
]
```

Figura D-9 - Reposta do pedido GetAllLocalizacoes

GetAllRedes

Este método serve para consultar todas as redes que uma empresa tem. Na figura D-10, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*.

<http://dominio.net/api/webservice/GetAllRedes/chave>

```
[
  - {
    id_rede: 1,
    id_empresa: 2,
    id_local: 1,
    IDRe: 100,
    Identificador: "1124124",
    nomerede: "Postos Transformação",
    longitude: "-8.407931599999984",
    latitude: "40.1959904",
    cidade: "Coimbra",
    localidade: "VF ",
    morada: "Vale das Flores",
    codigopostal: "3030"
  },
  - {
    id_rede: 2,
    id_empresa: 2,
    id_local: 1,
    IDRe: 300,
    Identificador: "092357",
    nomerede: "Posto de Transformação de Celas",
    longitude: "-8.461343499999998",
    latitude: "40.1991451",
    cidade: "Coimbra",
    localidade: null,
    morada: "Quinta de Voimarães",
    codigopostal: "3030-377"
  }
]
```

Figura D-10 - Resposta do pedido GetAllRedes

GetAllRedesByLocal

Este método serve para consultar todas as redes que uma empresa tem numa determinada localização. Na figura D-11, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso, e no último argumento do *url* o *id* da rede a consultar.

<http://dominio.net/api/webservice/GetAllRedesByLocal/chave/1>

```
[
  - {
    id_rede: 1,
    id_empresa: 2,
    id_local: 1,
    IDRe: 100,
    Identificador: "1124124",
    nomerede: "Postos Transformação",
    longitude: "-8.407931599999984",
    latitude: "40.1959904",
    cidade: "Coimbra",
    localidade: "VF ",
    morada: "Vale das Flores",
    codigopostal: "3030"
  },
  - {
    id_rede: 2,
    id_empresa: 2,
    id_local: 1,
    IDRe: 300,
    Identificador: "092357",
    nomerede: "Posto de Transformação de Celas",
    longitude: "-8.461343499999998",
    latitude: "40.1991451",
    cidade: "Coimbra",
    localidade: null,
    morada: "Quinta de Voimarães",
    codigopostal: "3030-377"
  }
]
```

Figura D-11 - Resposta do pedido GetAllRedesByLocal

GetAllGrupos

Este método serve para consultar todos os grupos de sensores de todas as redes de sensores de uma empresa. Na figura D-12, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*.

<http://dominio.net/api/webservice/GetAllGrupos/chave>

```
[
  - {
    id_grupo: 1,
    nomegrupo: "Circuito 1",
    descricao: "Circuito 1"
  },
  - {
    id_grupo: 2,
    nomegrupo: "Circuito 2",
    descricao: "Circuito 2"
  },
  - {
    id_grupo: 3,
    nomegrupo: "Circuito 3",
    descricao: "Circuito 3"
  },
  - {
    id_grupo: 4,
    nomegrupo: "Circuito 4",
    descricao: "Circuito 4"
  },
  - {
    id_grupo: 5
```

Figura D-12 - Resposta do pedido *GetAllGrupos*

GetSensoresByGrupo

Este método serve para consultar todos os sensores de um determinado grupo de uma rede de sensores de uma empresa. Na figura D-13, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave acesso ao *Web Service*, e no último argumento o *id* do grupo a consultar.

<http://dominio.net/api/webservice/GetSensoresByGrupo/chave/7>


```
[
  - {
    id_sensor: 20,
    id_empresa: 2,
    id_modelo: 7,
    id_rede: 1,
    id_grupo: 7,
    IDsensor: 70,
    nomesensor: "Temperatura",
    descricao: null
  },
  - {
    id_sensor: 21,
    id_empresa: 2,
    id_modelo: 7,
```

Figura D-13 - Resposta do pedido GetSensoresByGrupo

GetAllModelos

Este método serve para consultar todos os modelos de sensores que uma empresa tem nas suas redes de sensores. Na figura D-14, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*.

<http://dominio.net/api/webservice/GetAllModelos/chave>

```
[
  - {
    id_modelo: 1,
    nome: "EWS DTTI-reh"
  },
  - {
    id_modelo: 2,
    nome: "EWS DTF"
  },
  - {
    id_modelo: 3,
    nome: "EWS DTPump"
  },
  - {
    id_modelo: 4,
    nome: "EWS DTPD"
  },
  - {
    id_modelo: 5,
    nome: "EWS DTP"
  },
  - {
    id_modelo: 6,
    nome: "EWS DTP-g"
  },
  - {
    id_modelo: 7,
    nome: "EWS DTHT"
  },
]
```

Figura D-14 - Reposta ao pedido GetAllModelos

GetSensoresByModelo

Este método serve para consultar todos os sensores de um determinado modelo de uma rede de sensores de uma empresa. Na figura D-15, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*, e no último argumento o *id* do modelo a consultar.

<http://dominio.net/api/webservice/GetSensoresByModelo/chave/1>

```
[
  - {
    id_sensor: 1,
    id_empresa: 2,
    id_modelo: 1,
    id_rede: 1,
    id_grupo: 1,
    IDSensor: 10,
    nomesensor: "C1_L1",
    descricao: null
  },
  - {
    id_sensor: 2,
    id_empresa: 2,
    id_modelo: 1,
    id_rede: 1,
    id_grupo: 1,
    IDSensor: 11,
    nomesensor: "C1_L2",
    descricao: null
  },
  ,
]
```

Figura D-15 - Reposta ao pedido *GetSensoresByModelo*

GetAllEntradasBySensor

Este método serve para consultar todas as entradas de um determinado sensor de uma rede de sensores de uma empresa. Na figura D-16, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*, e no último argumento o *id* do sensor a consultar.

<http://dominio.net/api/webservice/GetAllEntradasBySensor/chave/10>

```
[
  - {
    id_entrada: 1,
    id_empresa: 2,
    codfuncao: 11,
    nome: "C1_L1",
    legendachart: "Corrente",
    descricao: "Corrente C1",
    valorMax: "2000",
    valorAlerta: "1000",
    valorMin: "0"
  }
]
```

Figura D-16 - Resposta ao pedido GetAllEntradasBySensor

GetAllMensagensBySensorEntrada

Este método serve para consultar todas as mensagens de uma entrada referente a um sensor de uma rede de sensores de uma empresa. Na figura D-17, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*, e no penúltimo argumento o *id* do sensor, e no último argumento o *id* da entrada a consultar.

<http://dominio.net/api/webservice/GetAllMensagensBySensorEntrada/chave/10/1>

```
[
  - {
    id_evento: 1,
    id_empresa: 2,
    id_entrada: 1,
    IDRede: 100,
    IDSensor: 10,
    codigofuncao: 11,
    valor: null,
    valor_argumento: 0,
    valor_extra: null,
    valor_ordem: null,
    data: "2013-07-09T11:44:23",
    sequencia: null
  },
  - {
    id_evento: 2,
    id_empresa: 2,
    id_entrada: 1,
    IDRede: 100,
    IDSensor: 10,
    codigofuncao: 11,
    valor: null,
    valor_argumento: 410,
    valor_extra: null,
    valor_ordem: null,
    data: "2013-07-09T13:49:20",
    sequencia: null
  },
]
```

Figura D-17 - Reposta ao pedido GetAllMensagensBySensorEntrada

GetNumLastMensagens

Este método serve para consultar as últimas mensagens de uma entrada referente a um sensor de uma rede de sensores de uma empresa. Na figura 18, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*, e no penúltimo argumento o *id* sensor, e no último argumento o número das últimas mensagens recebidas pelo sensor que se pretende consultar.

<http://dominio.net/api/webservice/GetNumLastMensagens/chave/10/2>

```
[
  - {
    id_evento: 56055,
    id_empresa: 2,
    id_entrada: 1,
    IDRede: 100,
    IDSensor: 10,
    codigofuncao: 11,
    valor: null,
    valor_argumento: 399,
    valor_extra: null,
    valor_ordem: null,
    data: "2014-09-01T01:51:37",
    sequencia: null
  },
  - {
    id_evento: 56054,
    id_empresa: 2,
    id_entrada: 1,
    IDRede: 100,
    IDSensor: 10,
    codigofuncao: 11,
    valor: null,
    valor_argumento: 440,
    valor_extra: null,
    valor_ordem: null,
    data: "2014-09-01T01:41:26",
    sequencia: null
  },
]
```

Figura D-18 - Reposta ao pedido *GetNumLastMensagens*

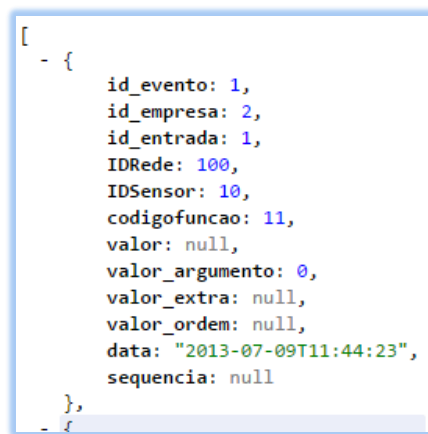
GetAllMensagens

Este método serve para consultar todas as mensagens da entrada referente a um sensor de uma rede de sensores de uma empresa. Na figura D-19, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*, e no último argumento o *id* do sensor a consultar.

<http://dominio.net/api/webservice/GetAllMensagens/chave/10>



```
[
  - {
    id_evento: 1,
    id_empresa: 2,
    id_entrada: 1,
    IDRede: 100,
    IDSensor: 10,
    codigofuncao: 11,
    valor: null,
    valor_argumento: 0,
    valor_extra: null,
    valor_ordem: null,
    data: "2013-07-09T11:44:23",
    sequencia: null
  },
  - {
```

Figura D-19 - Reposta ao pedido GetAllMensagens

GetAllMensagensAlarme

Este método serve para consultar todas as mensagens em alarme de uma entrada referente a um sensor de uma rede de sensores de uma empresa. Na figura D-20, vê-se o resultado obtido no pedido ao *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*, e no último argumento o *id* do sensor a consultar.

<http://dominio.net/api/webservice/GetAllMensagensAlarme/chave/10>

```
[
  - {
    id_evento: 1,
    id_empresa: 2,
    id_entrada: 1,
    IDRede: 100,
    IDSensor: 10,
    codigofuncao: "11",
    valor_lido: "0",
    valorMax: "L",
    valorAlerta: "L",
    valorMin: "0",
    descricaoalarme: "Corrente C1",
    data: "2014-09-01T03:39:09.707",
    sequencia: null,
    alarmeenviado: true,
    alarmetratado: false,
    Identificador: "1124124",
    username: null,
    operatordescricao: null,
    operatardate: null,
    periodo: null,
    tipoperiodo: null
  },
]
```

Figura D-20 - Reposta ao pedido GetAllMensagensAlarme

GetAllMensagensErro

Este método serve para consultar todas as mensagens de erro das várias redes de sensores de uma empresa. Na figura D-21, vê-se o resultado obtido a partir do *Web Service* no formato *JSON*.

Exemplo do pedido

No *link* após o método a consultar, tem-se de definir a chave de acesso ao *Web Service*, e no último argumento o *id* do sensor a consultar.

<http://dominio.net/api/webservice/GetAllMensagensErro/chave>

```
[
  - {
    id_erro: 1,
    IDRede: 100,
    IDSensor: null,
    valor_lido: "45861",
    descricaoErro: "Sensor não configurado na rede id: 100 ou Sensor: 10 com código função mal definido: 232",
    data: "2014-07-27T02:02:01.027",
    id_empresa: 2,
    errovisto: false
  },
  - {
    id_erro: 2,
    IDRede: 100,
    IDSensor: null,
    valor_lido: "45861",
    descricaoErro: "Sensor não configurado na rede id: 100 ou Sensor: 10 com código função mal definido: 232",
    data: "2014-07-27T02:05:00.177",
    id_empresa: 2,
    errovisto: false
  },
]
```

Figura D-21 - Reposta ao pedido GetAllMensagensErro

PostMensagem

Este método serve para submissão de mensagens por parte da rede de sensores para o *Web Service*. Na figura D-22 apresenta-se o exemplo da construção do objeto no formato *JSON*, para submissão dos dados para o *Web Service*.

Exemplo do pedido

No *link* v método a submeter dados no formato *JSON*, tem-se de definir a chave de acesso ao *Web Service*, para o pedido *POST* ser aceite.

<http://dominio.net/api/webservice/PostMensagem/chave>

```
Content-Type: application/json
Host: dominio.net
{
  id_empresa: 3,
  id_entrada: 1,
  IDRede: 200,
  IDSensor: 20,
  codigofuncao: 12,
  valor: 1560,
  valor_argumento: 1560,
  valor_extra: 0,
  valor_ordem: 0,
  data: "2014-08-29T18:22:00.000",
}
```

Figura D-22 - Exemplo de construção do objeto no formato JSON para ser feito o pedido PostMensagem

4. MANUAL UTILIZADOR APLICAÇÃO SCADA WEB

Nesta secção apresenta-se o manual de utilizador da aplicação *SCADA Web*. Esta aplicação é usada pelo administrador da *eneidaws* (*AdministradorGeral*), pelos clientes da *eneidaws* (*ClienteAdministrador*) e pelos funcionários dos clientes da *eneidaws* (*ClienteUtilizador*). Foram definidas três categorias de utilizadores (*AdministradorGeral*, *ClienteAdministrador* e *ClienteUtilizador*), para representar os privilégios e regras de acesso à aplicação *SCADA Web*. Com base nestas regras de acesso, a informação apresentada ao utilizador varia, ficando disponíveis mais ou menos funcionalidades e opções. Esta aplicação tem como finalidade a configuração das redes de sensores e a monitorização, o controlo e a análise das redes configuradas.

Página Login

O utilizador, quando acede pelo *browser* à aplicação *SCADA Web*, tem disponíveis as seguintes ações: fazer *login* na aplicação, recuperar *password* ou ver os contactos da *eneidaws*.

Para poder usar a aplicação *SCADA Web*, o utilizador tem que ter uma conta de *login* definida com a respetiva regra de acesso, onde deve introduzir o seu *username* e *password*, como se vê na figura D-23 para aceder à aplicação. Sempre que o utilizador não se lembre da sua *password*, pode recuperar a mesma na página vista na figura D-24, sendo gerada uma nova *password* enviada por *e-mail*. Na figura D-25, o utilizador pode ver os contactos da *eneidaws*.

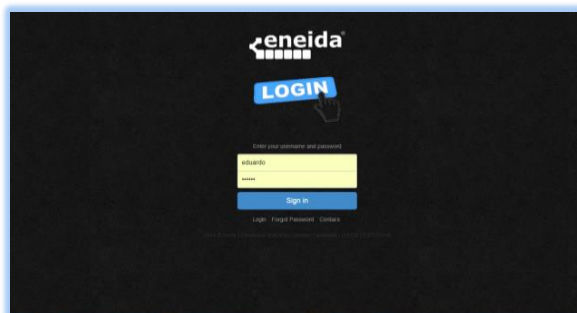


Figura D-23 - Página de Login

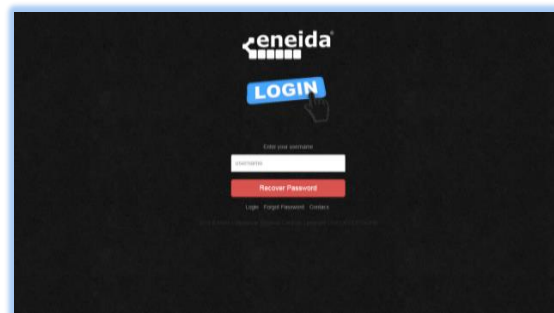


Figura D-24 - Página de Recuperar Password

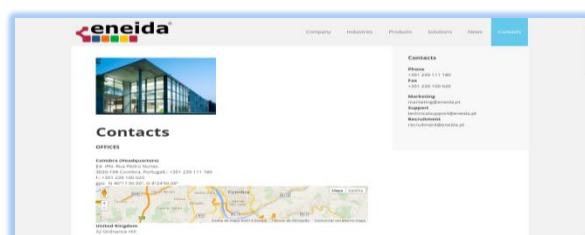


Figura D-25 - Página de Contactos da eneidaws

Página Home

Após o *login*, o utilizador tem acesso à página *home*. Nesta página, o utilizador tem acesso à opção *dashboard*, com acesso aos alarmes existentes nas redes de sensores e *links* de acesso aos diferentes menus. Do lado esquerdo encontram-se os vários menus com as diferentes opções vistas nas figuras D-26, D-27 e D-28. Esta informação disponível na página *home* varia de acordo com os níveis de privilégios dos diferentes utilizadores, como se pode ver nas figuras D-26, D-27 e D-28.

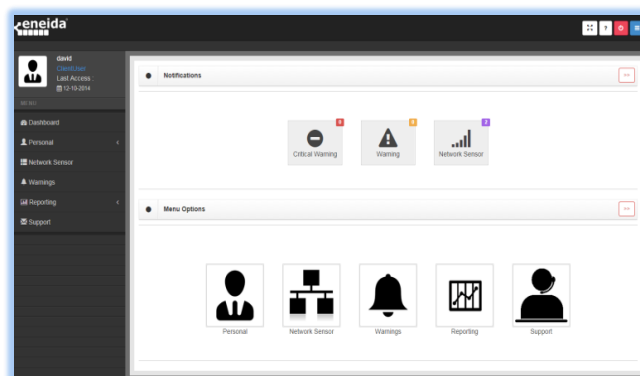


Figura D-26 - Página Home com as opções visíveis para o utilizador ClienteUtilizador

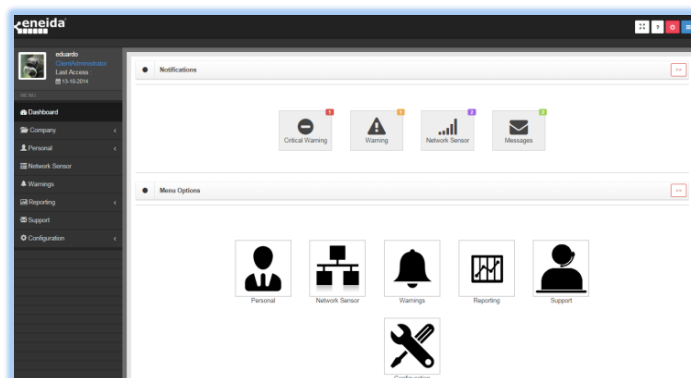


Figura D-27 - Página Home com as opções visíveis para o utilizador ClienteAdministrador

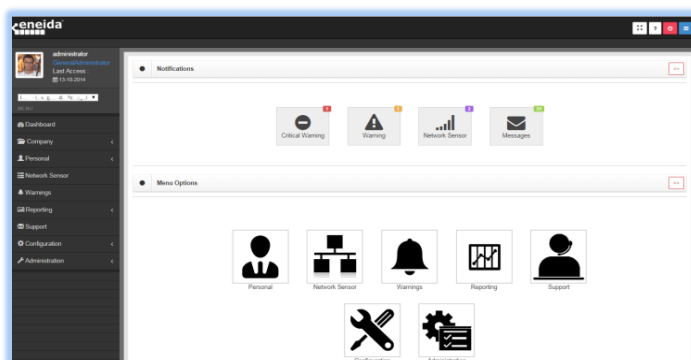


Figura D-28 - Página Home com as opções visíveis para o utilizador AdministradorGeral

Na página *home* e nas restantes páginas, qualquer utilizador tem acesso à opção *Ajuda*. Ao clicar no botão com o ícone ponto de interrogação, visto no canto superior direito da figura D-29, o utilizador poderá consultar ajuda referente às várias opções existentes na aplicação, com recurso a vídeos. Para sair da aplicação terminando a sessão, o utilizador pode fazê-lo clicando no botão vermelho no canto superior direito, visto na figura D-29. O botão azul permite alterar a visualização do menu do lado esquerdo. O primeiro botão ao lado do botão ponto de interrogação permite ao utilizador colocar a visualização do *site* em *fullscreen*.

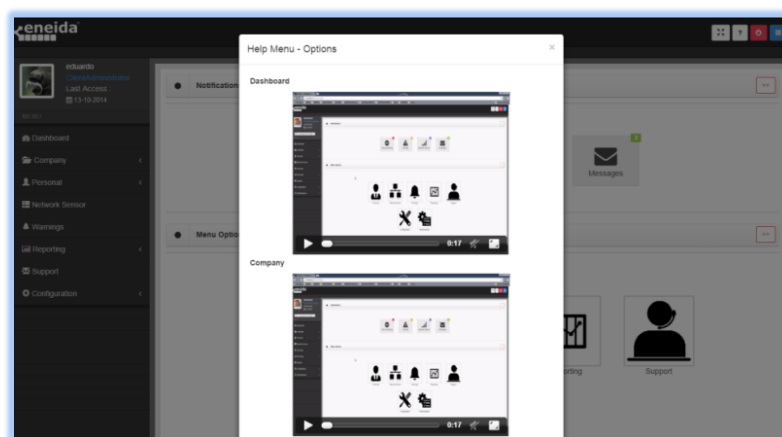


Figura D-29 - Página ajuda disponível para os utilizadores

Página Home, opção do Menu *Company*

A opção do menu *Company* permite ao utilizador *ClienteAdministrador* visualizar informação da sua empresa criada na aplicação *Web*, pelo utilizador administrador da *eneidaws*. Na figura D-30, podem ver-se os dados da empresa e a possibilidade de o utilizador fazer *upload* do logotipo da sua empresa.

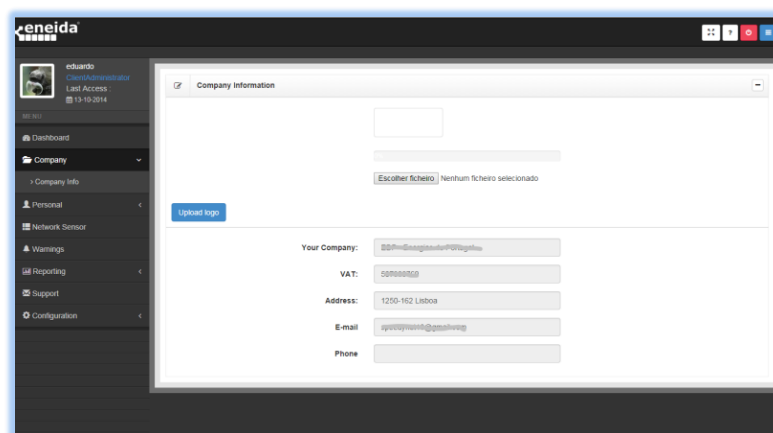


Figura D-30 - Página company com informação disponível sobre os dados da empresa

Página Home, opção do Menu *Personal*

A opção do menu *Personal* permite ao utilizador *ClienteAdministrador* e ao *ClienteUtilizador* consultar a sua informação pessoal. Na página vista na figura D-31, o utilizador pode alterar a sua foto pessoal, nome e *e-mail*. Pode ainda ver que a sua conta de utilizador está atribuída a uma determinada empresa e, na página vista na figura D-32, o utilizador pode alterar a sua *password* sempre que achar conveniente.

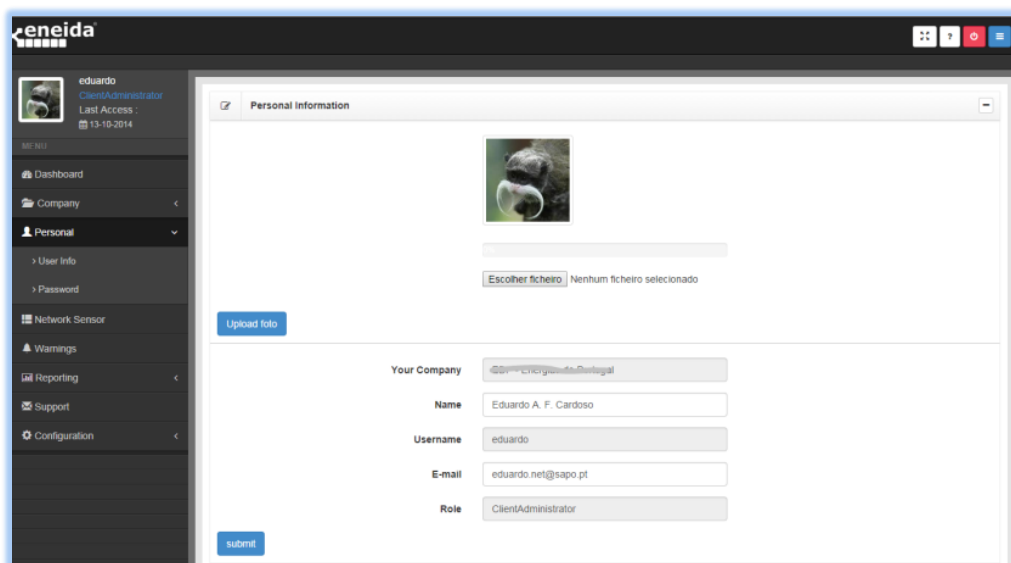


Figura D-31 - Página Informação Pessoal Utilizador

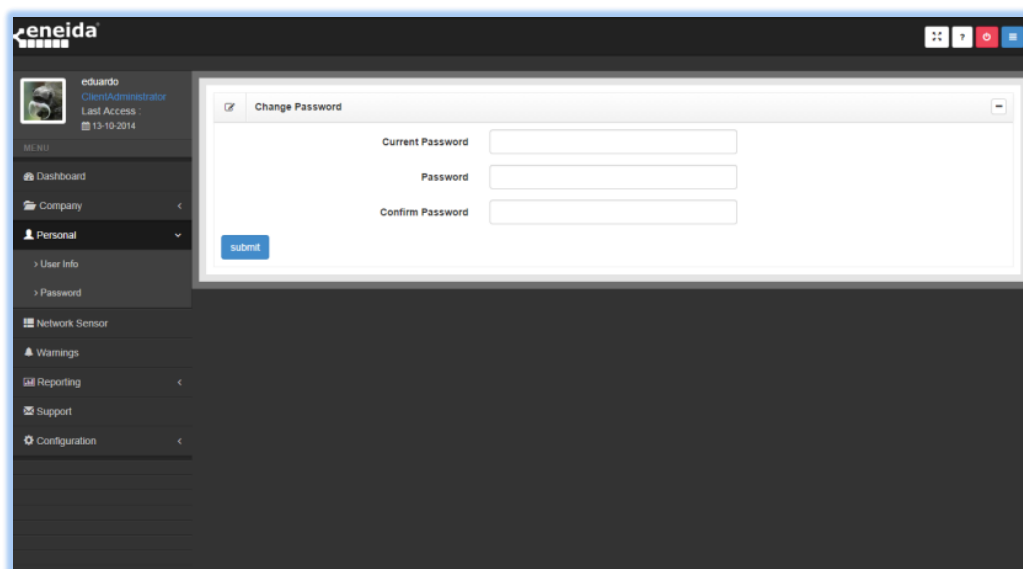


Figura D-32- Página para utilizador alterar a sua password

Página Home, opção do Menu *Network Sensor*

A opção do menu *Network Sensor* permite ao utilizador *ClienteAdministrador* e ao utilizador *ClienteUtilizador* visualizar as redes de sensores configuradas no mapa, com recurso às coordenadas *GPS* que cada rede tem. Os pontos das redes vistos no mapa podem assumir três cores. A cor verde significa que não existe nenhum alarme nessa rede, a amarela significa que existe um alerta para alarme, ficando esta rede sob alerta, e a cor vermelha significa que existe um alarme crítico. Na figura D-33, podemos ver a página com a vista das redes de sensores no mapa; caso o utilizador prefira pode ver as redes de sensores no formato tabela, visto na figura D-34.

O utilizador tem a possibilidade de pesquisar pelo código único definido para cada rede, ou, se preferir pode clicar nos botões vermelho, amarelo e verde para visualizar só alarmes críticos, alerta de alarmes ou redes sem alarmes. O botão azul serve para visualizar todas as redes independentemente do seu estado (em alarme ou não). Como se pode ver na figura D-33, sempre que o utilizador clicar num ponto pode ver informação resumida sobre a rede de sensores; caso este pretenda, pode ver a vista da rede de sensores clicando no *link* “+ info network”. Ao clicar neste *link*, o utilizador será redirecionado para a vista da rede de sensores com os últimos valores adquiridos nessa rede, como se pode ver na figura D-35. A rede escolhida para este exemplo tem um alerta para alarme de cor amarela: pode-se ver em detalhe, na figura D-35, que o problema está no sensor que obtém dados do circuito 1 e linha R.

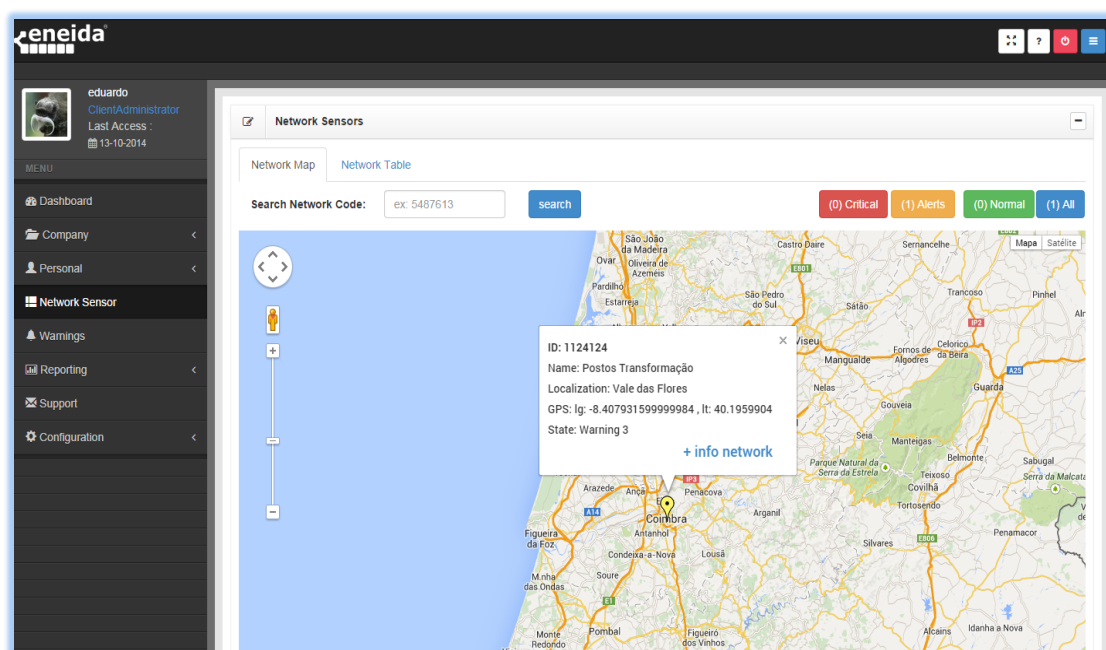


Figura D-33 - Página com mapa onde se veem as redes de sensores configuradas

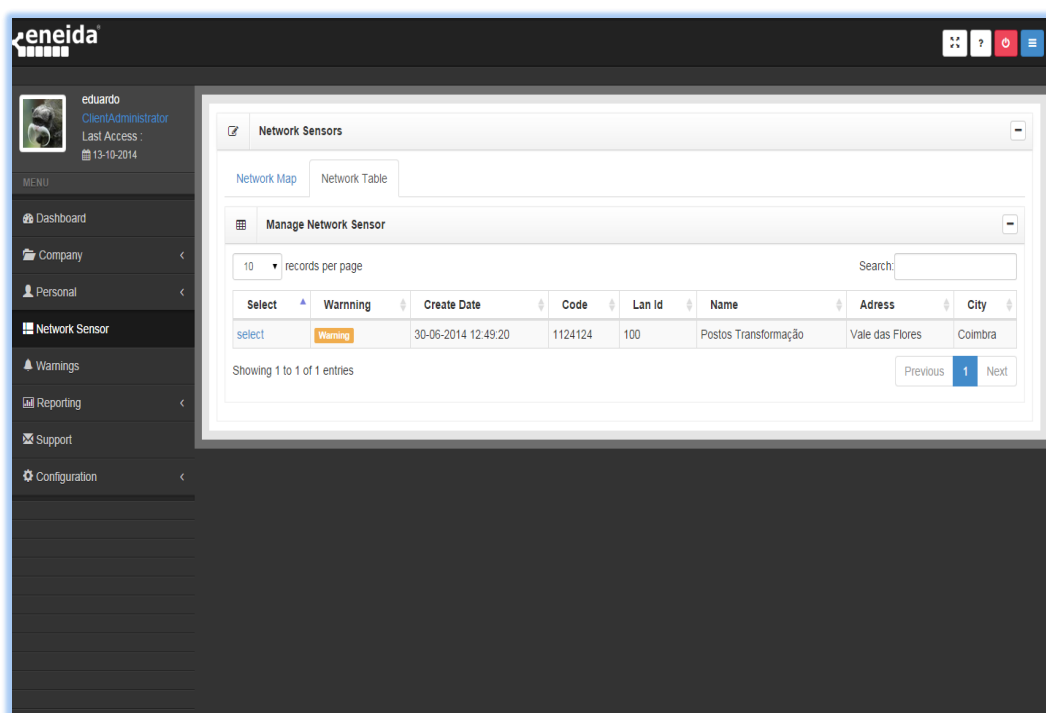


Figura D-34 - Página com vista das redes de sensores configuradas no formato tabela

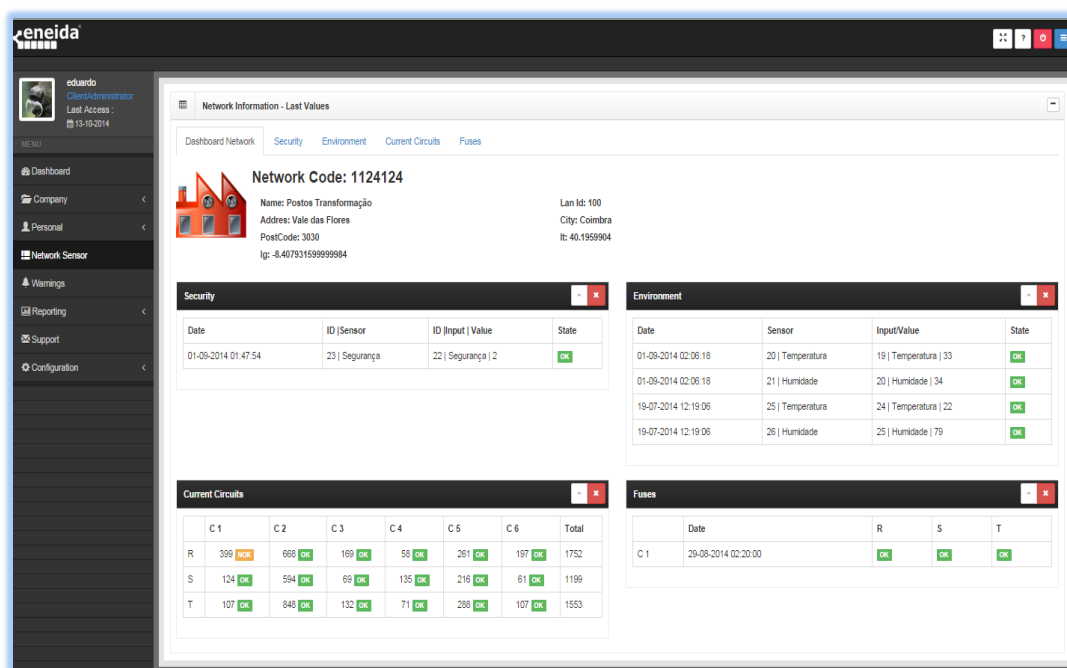


Figura D-35 - Página com a vista dos últimos valores adquiridos pela rede de sensores e seu estado

Página Home, opção do Menu Warnings

A opção do menu *Warnings* permite ao utilizador *ClienteAdministrador* e ao utilizador *ClienteUtilizador* visualizar todos os alarmes críticos, como se vê na figura D-36, e alarmes não críticos (alertas), vistos na figura D-37. Pode ver-se em detalhe a descrição da ocorrência que gerou o alarme. Especificamente para o utilizador *ClienteAdministrador*, existe a possibilidade de definir um alarme como resolvido deixando uma pequena descrição, saindo estes alarmes da lista. Para o utilizador *ClienteAdministrador* existe ainda a possibilidade de definir quais os utilizadores da sua empresa que recebem alarmes referentes às redes de sensores, como se pode ver na figura D-38.

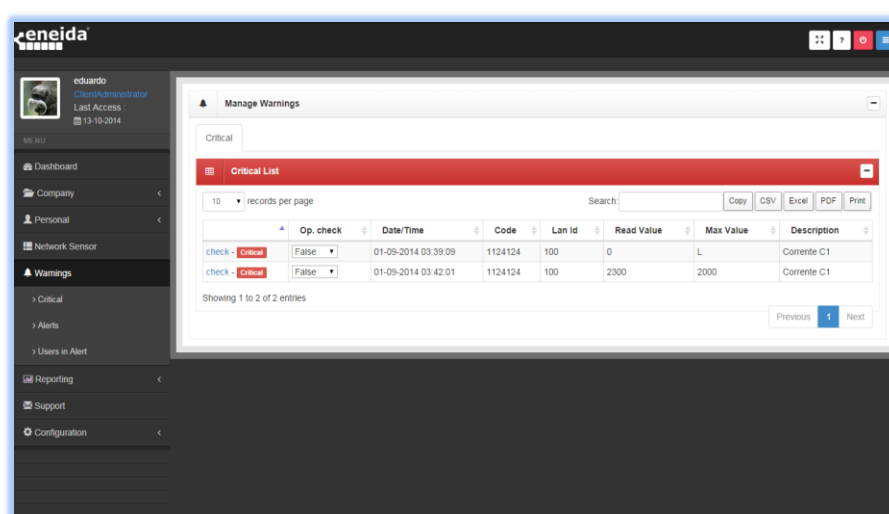


Figura D-36 - Página com lista de alarmes críticos

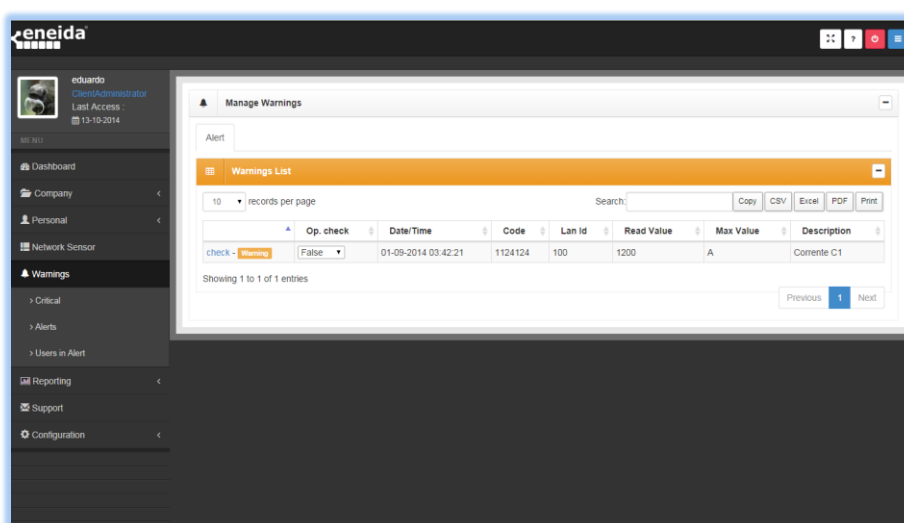


Figura D-37 - Página com lista de alarmes não críticos (alertas)

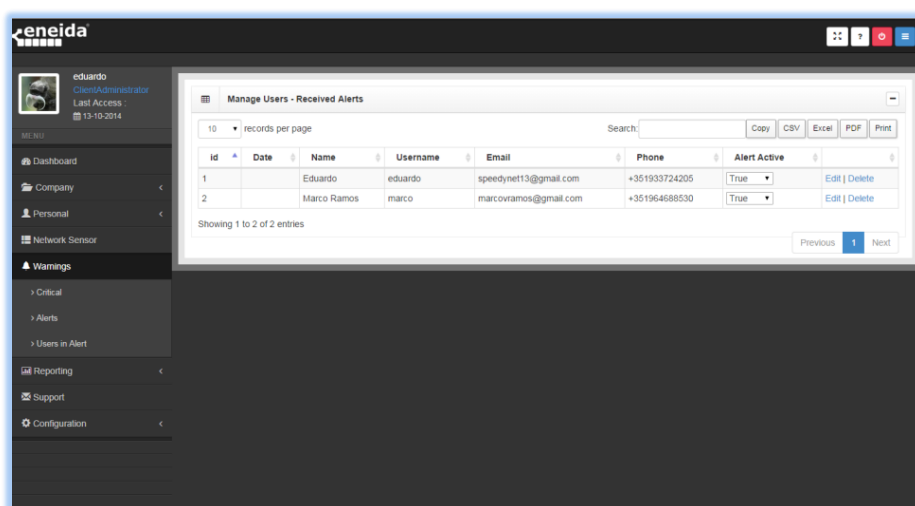


Figura D-38 - Página com lista de utilizadores que recebem alarmes na ocorrência de alarmes

Página Home, opção do Menu *Reporting*

A opção do menu *Reporting* permite aos utilizadores visualizar, monitorizar e analisar toda a informação sobre as suas redes de sensores, através do histórico de dados, no formato gráfico, tabelas e por pesquisas customizadas de informação. Esta opção só está disponível para os clientes da *eneidaws* que tenham a licença para acesso completo a todas a funcionalidades e opções da aplicação *Web*. Na figura D-39, podem ver-se os submenus desta opção, onde é possível consultar o histórico de dados dos sensores das redes de sensores, o histórico de dados de alarmes ocorridos nas redes de sensores e, na última opção, informação relativa às redes de sensores. Na figura D-39, pode ver-se a lista de sensores de uma rede selecionada.

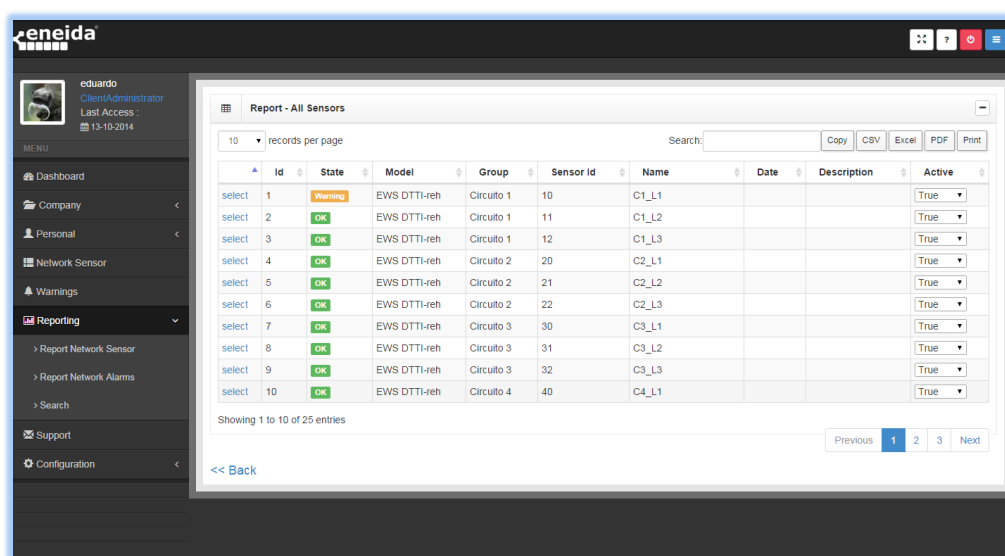


Figura D-39 - Página com lista de sensores associados a uma rede

Nesta lista da figura D-39, podem ser vistos todos os sensores e o seu estado (em alarme ou não). Ao selecionar um sensor pode ver-se o histórico de dados em gráficos ou tabelas. O utilizador pode analisar a informação do histórico de dados, entre uma data inicial e uma data final, como visto na figura D-40, existindo ainda a possibilidade de o utilizador exportar os dados do gráfico gerado para ficheiro Excel. Na figura D-41, o utilizador pode ver o histórico de dados do sensor em tabela, entre uma data inicial e uma data final, e exportar de igual forma a informação para Excel, entre outros.

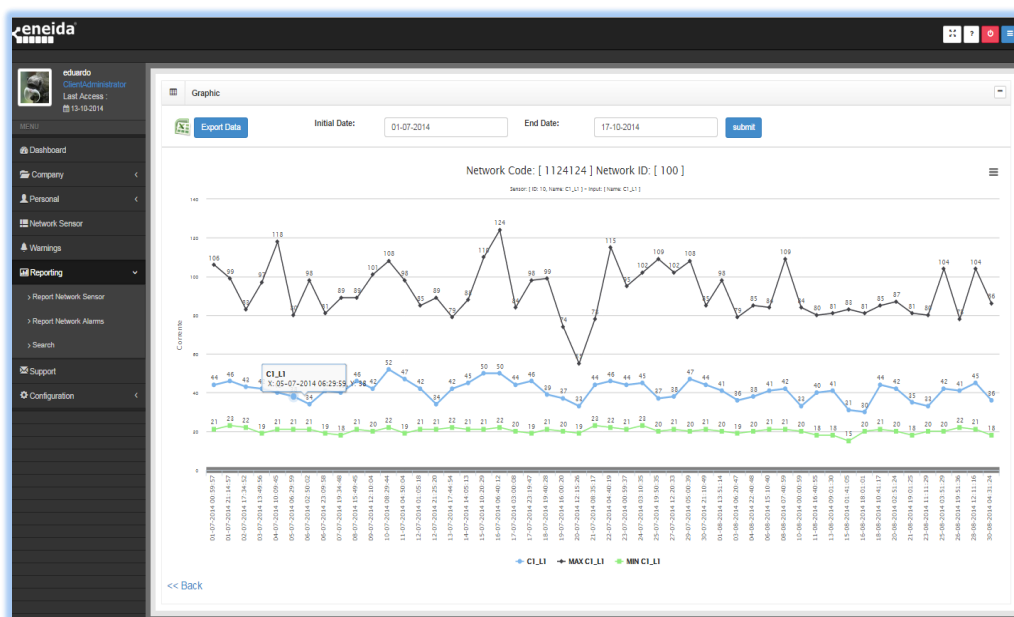


Figura D-40 - Página com gráfico referente ao histórico de dados de um sensor

Date	R	S	T
29-08-2014 02:20:00	OK	OK	OK
29-08-2014 01:19:59	OK	OK	OK
29-08-2014 00:19:57	OK	OK	OK
28-08-2014 23:19:57	OK	OK	OK
28-08-2014 22:19:56	OK	OK	OK
28-08-2014 21:19:53	OK	OK	OK
28-08-2014 06:16:56	NOK	NOK	NOK
28-08-2014 05:20:59	OK	OK	OK
28-08-2014 04:21:13	OK	OK	OK
28-08-2014 03:21:09	OK	OK	OK

Figura D-41 - Página com a tabela referente ao histórico de dados de um sensor

Na figura D-42, pode ver-se a página para o utilizador proceder a pesquisas customizadas. Nesta página o utilizador pode pesquisar informação referente às redes, sensores, grupo de sensores e respetivas entradas dos sensores, entre uma data inicial e uma data final. Posteriormente, o utilizador pode exportar o resultado da pesquisada para Excel, *PDF*, entre outros formatos. Na página de pesquisa da figura D-43, o utilizador pode fazer o mesmo tipo de pesquisa que na página da figura D-42, com a particularidade de poder pesquisar apenas por alarmes críticos ou alertas para alarmes ou por ambos.

Date	local	Net. code	Net. Id	Sensor Id	Sensor Name	Input Id	Input Name	Value
01-07-2014 00:59:57	Coimbra	1124124	100	10	C1_L1	1	C1_L1	356
01-07-2014 01:04:54	Coimbra	1124124	100	10	C1_L1	1	C1_L1	333
01-07-2014 01:09:54	Coimbra	1124124	100	10	C1_L1	1	C1_L1	309
01-07-2014 01:14:53	Coimbra	1124124	100	10	C1_L1	1	C1_L1	286
01-07-2014 01:19:49	Coimbra	1124124	100	10	C1_L1	1	C1_L1	306
01-07-2014 01:24:49	Coimbra	1124124	100	10	C1_L1	1	C1_L1	293
01-07-2014 01:29:46	Coimbra	1124124	100	10	C1_L1	1	C1_L1	264
01-07-2014 01:34:44	Coimbra	1124124	100	10	C1_L1	1	C1_L1	271
01-07-2014 01:39:41	Coimbra	1124124	100	10	C1_L1	1	C1_L1	285
01-07-2014 01:44:43	Coimbra	1124124	100	10	C1_L1	1	C1_L1	270

Figura D-42 - Página pesquisa de informação das redes de sensores customizada

State	Op. check	Date/Time	Code	Lan Id	Read Value	Max Value	Description
No data available in table							

Figura D-43 - Página pesquisa de informação das redes de sensores customizada por alarmes ocorridos

Na figura D-44, apresenta-se outra página de pesquisa em que o utilizador pode fazer as seguintes pesquisas: pesquisar por todos os sensores que nunca comunicaram com o sistema SCADA, ou então pesquisar por todos os sensores que não comunicam, por exemplo, há mais de 10 (segundos, minutos, horas ou dias), ou pesquisar por todos os sensores que não comunicam, por exemplo, há menos de 10 (segundos, minutos, horas ou dias). Todas estas pesquisas que o utilizador tem à sua disposição, permitem que este tenha controlo sobre a aplicação e a informação armazenada. Com base nestas pesquisas o utilizador pode tirar conclusões sobre problemas que possam existir nas configurações realizadas na aplicação SCADA ou problemas existentes nas várias redes de sensores instaladas no terreno, entre outros problemas. O utilizador poderá, com todas estas pesquisas realizadas, exportar informação para formato Excel, *PDF*, entre outros.

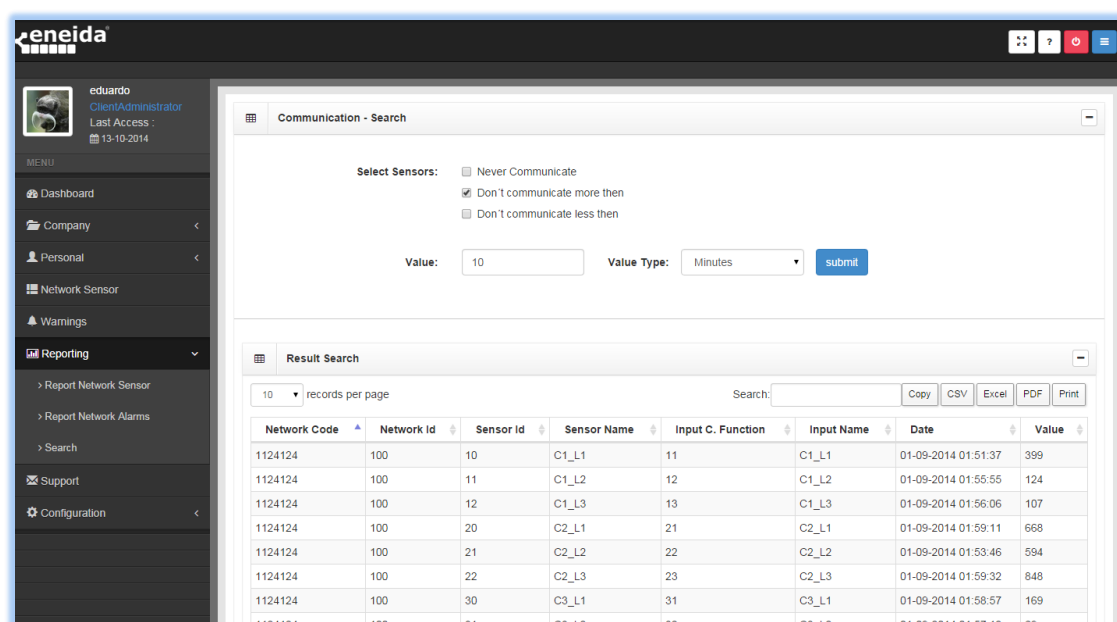


Figura D-44 - Página de pesquisa dos sensores que não comunicam há mais de determinado tempo

Página Home, opção do Menu Support

A opção do menu *Support* permite aos utilizadores usar esta opção para solicitar ajuda aos administradores relativamente a dúvidas ou problemas existentes na aplicação. Esta opção serve ainda para os administradores responderem aos pedidos de suporte dos utilizadores.

- Se o utilizador tiver o privilégio de acesso à aplicação como *ClienteUtilizador*, terá a página suporte vista na figura D-45, com a possibilidade de criar mensagens que serão enviadas diretamente para o *ClienteAdministrador*. As respostas do *ClienteAdministrador* são enviadas para o e-mail do *ClienteUtilizador*.
- Se o utilizador tiver o privilégio de acesso à aplicação como *ClienteAdministrador*, terá a página suporte vista na figura D-46, com a possibilidade de criar mensagens que serão diretamente enviadas para o *AdministradorGeral* e com a possibilidade de ver as

mensagens enviadas pelos seus utilizadores para si. As respostas do *AdministradorGeral* são enviadas para o e-mail do *ClienteAdministrador*.

- Se o utilizador tiver o privilégio de acesso à aplicação como *AdministradorGeral*, terá a página suporte vista na figura D-47, com a possibilidade de criar mensagens que serão diretamente enviadas para todos os *ClientesAdministradores* e a possibilidade de ver as mensagens enviadas pelos *ClientesAdministradores* com pedidos de suporte. As respostas do *AdministradorGeral* são enviadas para o e-mail aos *ClientesAdministradores*.

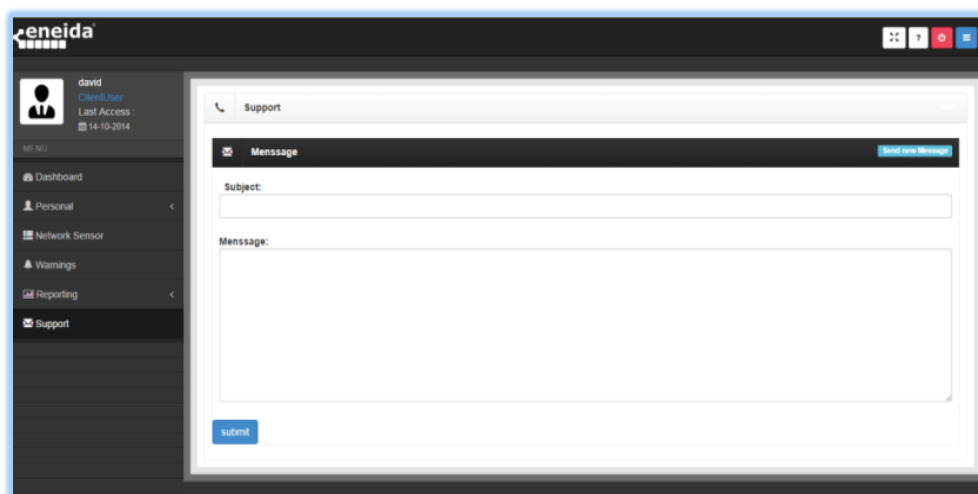


Figura D-45 - Página para criar mensagens de ajuda de suporte à aplicação

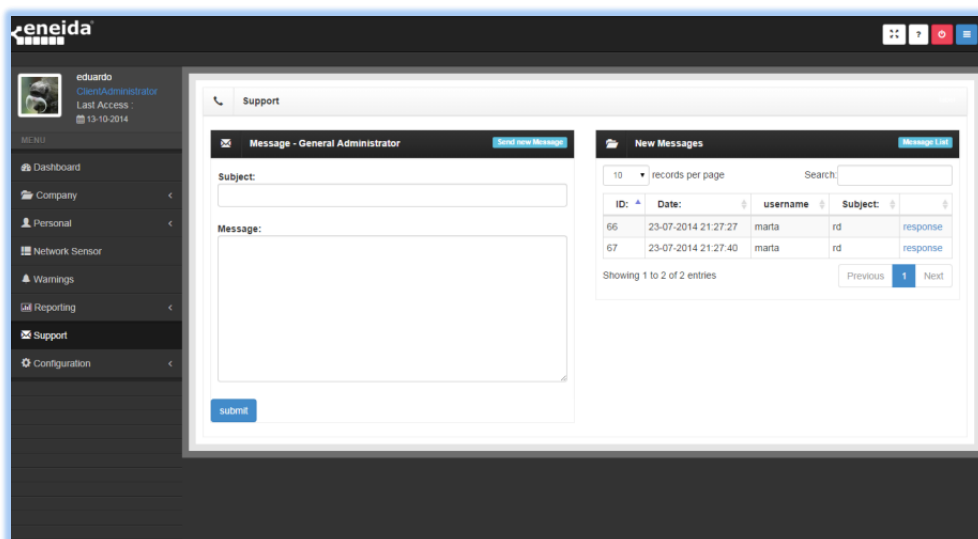


Figura D-46 - Página para criar mensagens de ajuda de suporte à aplicação e leitura de mensagens

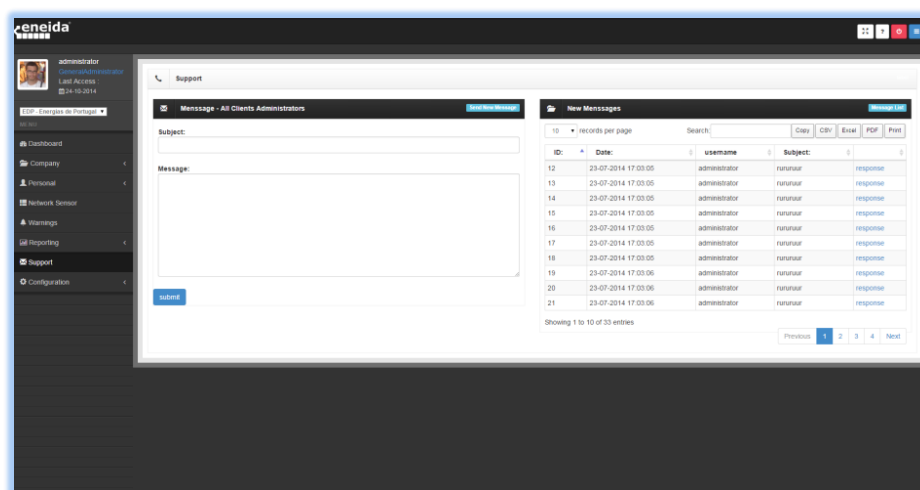


Figura D-47 -Página para criar mensagens de suporte à aplicação e leitura de mensagens

Página Home, opção do Menu Configuration

A opção do menu *Configuration* permite ao utilizador Cliente Administrador acesso às seguintes opções:

- Configurar Redes de Sensores
- Configurar Utilizadores
- Configurar Modelos e Grupos

Para criar uma rede de sensores, o utilizador tem um *Wizard*, que segue os seguintes passos: criar localização (figura D-48); criar rede (figura D-49) e configurar sensores e as entradas de cada sensor (figura D-50); e finalizar a configuração (figura D-51).

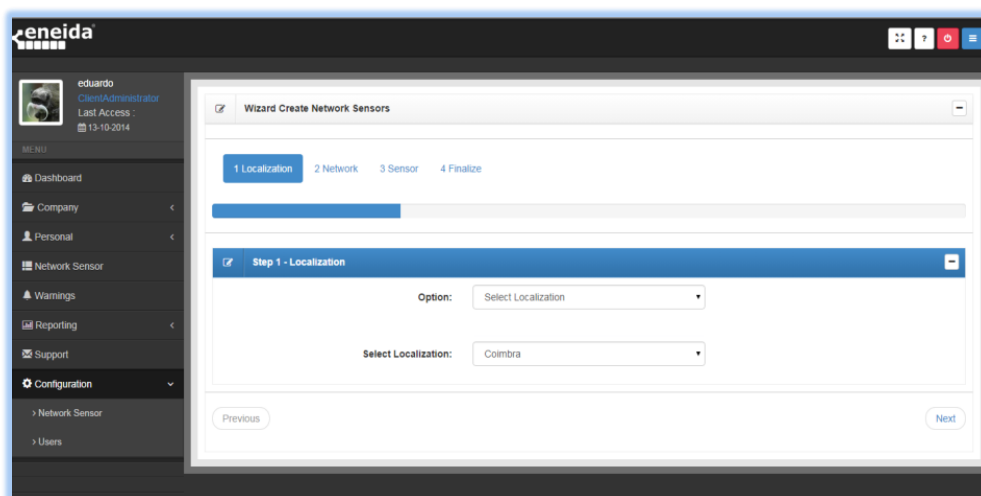


Figura D-48- Página Wizard criar localização

Client Administrator
Last Access : 13-10-2014

Dashboard
Company
Personal
Network Sensor
Warnings
Reporting
Support
Configuration
Network Sensor
Users

Wizard Create Network Sensors

1 Localization 2 Network 3 Sensor 4 Finalize

Step 2 - Network

Option: Create Network

Add Network Configuration

Image: Escolher ficheiro Nenhum ficheiro seleccionado

Code Network: 092357

ID Network: 300

Name: Posto de Transformação de Celas

City: Coimbra

Address: Quinta de Volmarães

Postcode: 3030-377

Search Coordinates

Figura D-49 - Página Wizard criar rede de sensores

Client Administrator
Last Access : 13-10-2014

Dashboard
Company
Personal
Network Sensor
Warnings
Reporting
Support
Configuration
Network Sensor
Users

Wizard Create Network Sensors

1 Localization 2 Network 3 Sensor 4 Finalize

Step 3 - Sensor

Group Sensor: Circuito 1

Sensor Model: EWS DTTI-reh

Sensor Configuration

ID Sensor: 300

Name: Sensor de Temperatura

Description: Sensor instalado dentro do PT

Add Inputs

Add row

Input	Value	Unit
E1 Temperatura	23	50 Minutes

Active Alerts: Temperatura E1

Entrada 1 - Sensor ID 300

Figura 50 - Página Wizard criar sensores e entradas de cada sensor

Client Administrator
Last Access : 13-10-2014

Dashboard
Company
Personal
Network Sensor
Warnings
Reporting
Support
Configuration
Network Sensor
Users

Wizard Create Network Sensors

1 Localization 2 Network 3 Sensor 4 Finalize

100%

Step 4 - Finalize

Create Network Success!

Previous Next

Figura D-51 - Página Wizard finalizado, rede criada com sucesso

Para criar várias redes de sensores de uma forma automática na aplicação *Web*, o utilizador pode importar uma lista de redes num ficheiro com o formato em *CSV*. O utilizador poderá também exportar todas as redes configuradas para um ficheiro no formato *CSV*.

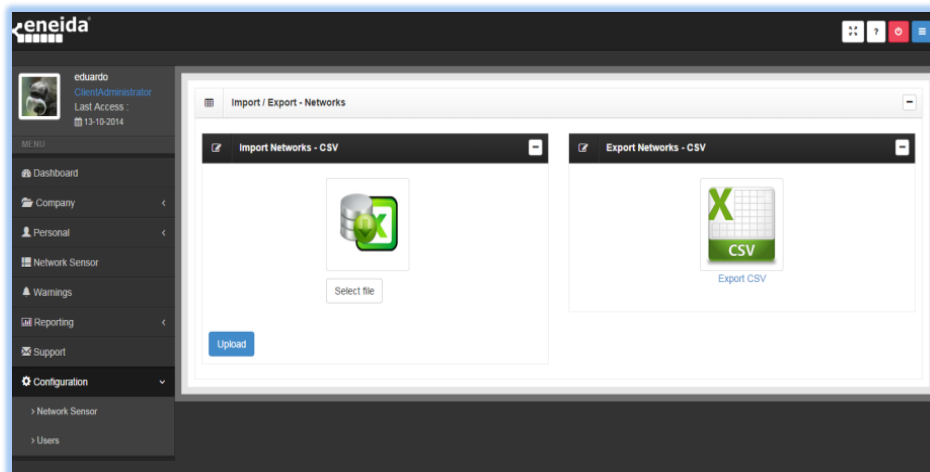


Figura D-52 - Página para importar ou exportar redes de sensores

Na figura D-53 apresenta-se a página para listagem de todas as redes de sensores configuradas. Aqui o utilizador pode ver todas as redes configuradas e, se o pretender, pode atualizar informação das redes de sensores (por exemplo, adicionando mais sensores), desativar redes, entre outros. As redes de sensores nunca são apagadas caso alguns sensores tenham registos de leituras efetuadas no terreno. Esta informação fica sempre guardada em base de dados, podendo o utilizador, na opção do menu *reporte*, ver o histórico dos dados das redes desativadas.

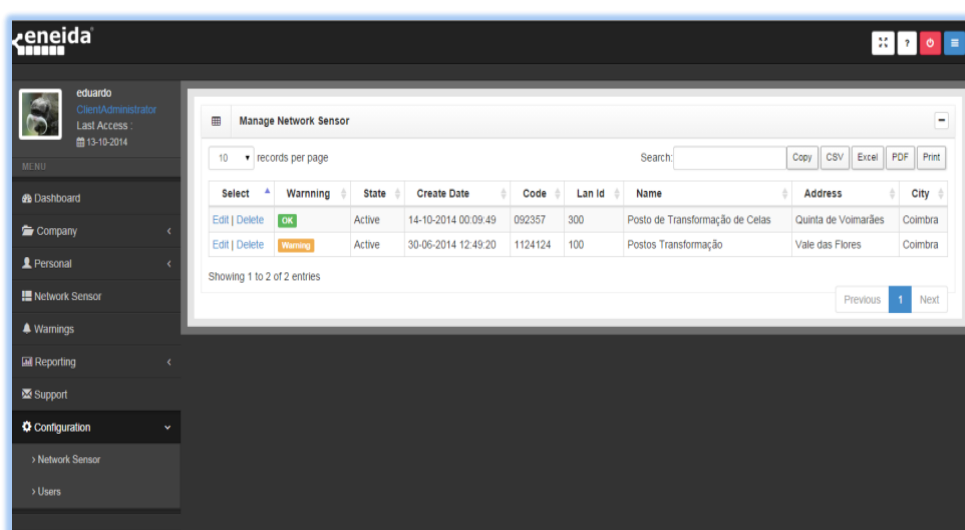


Figura D-53 - Página com lista das redes de sensores configuradas

Na figura D-54 apresenta-se a página com a listagem de todos os erros reportados pelo servidor relativo às redes de sensores. Por exemplo, se o utilizador tiver configurado na aplicação *Web* os sensores com *ids* únicos diferentes dos *ids* únicos dos sensores reais instalados no terreno, o servidor, ao receber uma mensagem não verificando correspondência com o que está configurado na aplicação *Web*, gera *log* de erros com mensagens que permitem ao utilizador perceber, por exemplo, por que razão não está a receber informações sobre determinada rede, resolvendo o problema de forma imediata.

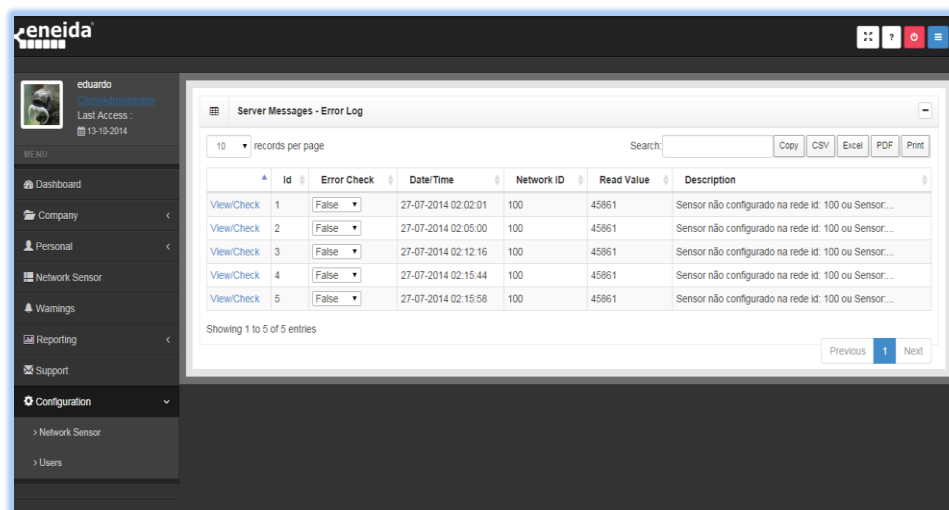


Figura D-54 - Página erros gerados pelo servidor

Na figura D-55 é apresentada a página para o utilizador *ClienteAdministrador* criar os seus utilizadores que são os funcionários da sua empresa. Na figura D-56 é apresentada a página para listar todos os utilizadores que tem configurados associados à sua empresa na aplicação *Web*. Na figura D-57 é apresentada a página com lista de todos os utilizadores que recebem alarmes. Na listagem de utilizadores, o *ClienteAdministrador* pode seleccionar quais dos seus funcionários recebem alarmes.

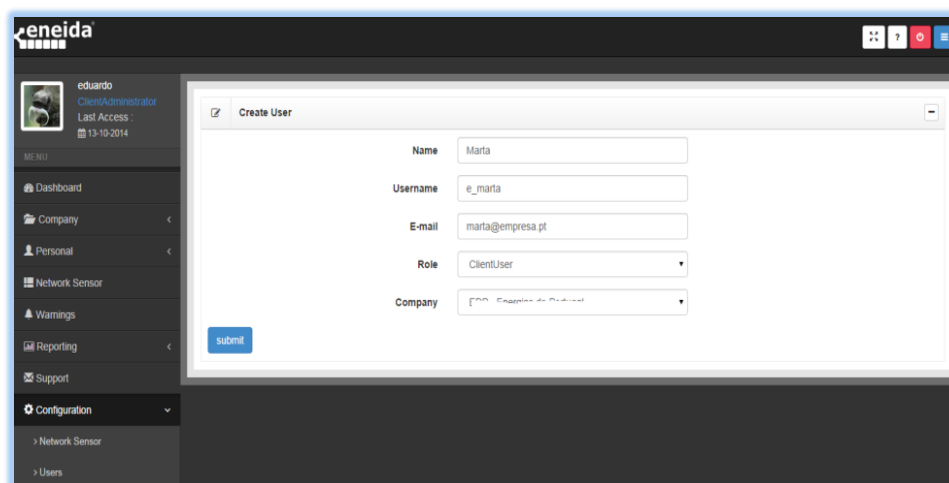


Figura D-55 - Página para criar utilizadores

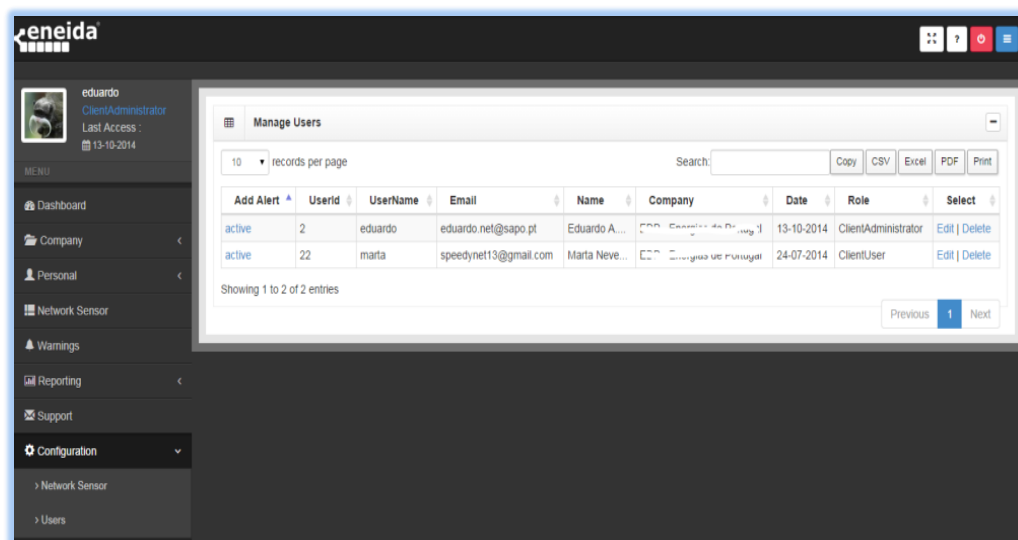


Figura D-56 - Página para listar todos os utilizadores criados pelo cliente administrador

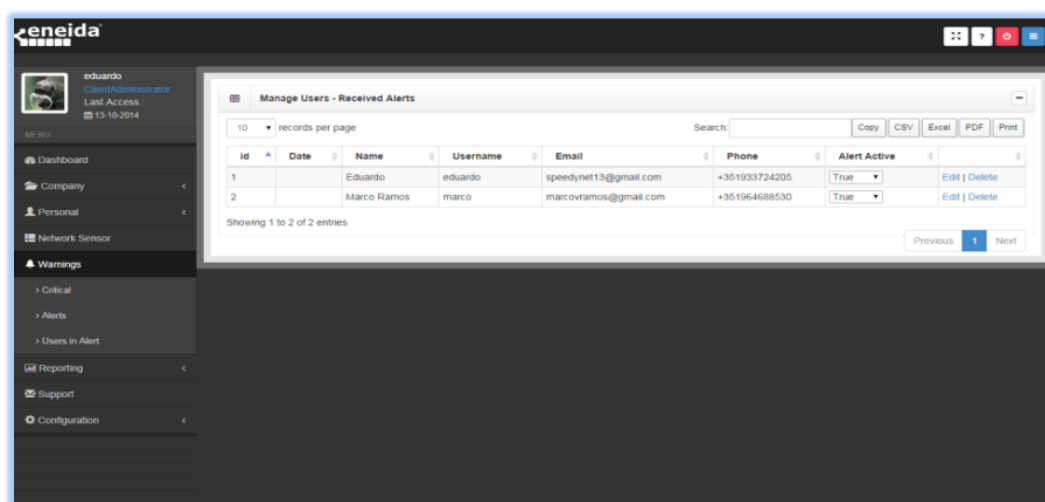
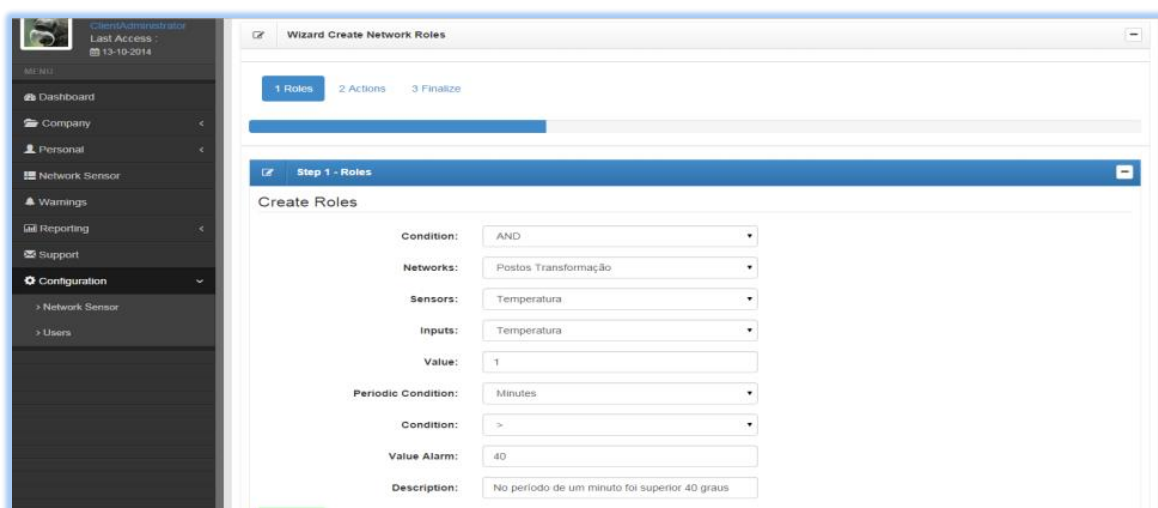


Figura D-57 - Página para listar todos os utilizadores que estão configurados para receber alarmes

Para criar regras e ações para alarmes é necessário configurar um *wizard* que segue os seguintes passos: (figura D-58) é apresentada a página para criar regras para determinadas redes de sensores e sensores específicos; (figura D-59) é apresentada a página para criar as ações que podem ser configuradas para determinada regra que seja despoletada. Na figura D-60 é apresentado o resultado após a configuração finalizada das regras e ações com sucesso. Por exemplo, se um sensor de uma rede e outro sensor de outra rede atingirem um certo valor, poderá ser despoletada a ação desativar todo o sistema e o envio de *SMS* e *e-mail* para o utilizador. Poderão também ser criadas regras que não necessitem da existência de ações específicas, pois por defeito são sempre enviados *e-mails* e *SMS*, na ocorrência de alarmes.



Client Administrator
Last Access : 13-10-2014

MENU

- Dashboard
- Company
- Personal
- Network Sensor
- Warnings
- Reporting
- Support
- Configuration
 - Network Sensor
 - Users

Wizard Create Network Roles

1 Roles 2 Actions 3 Finalize

Step 1 - Roles

Create Roles

Condition: AND

Networks: Postos Transformação

Sensors: Temperatura

Inputs: Temperatura

Value: 1

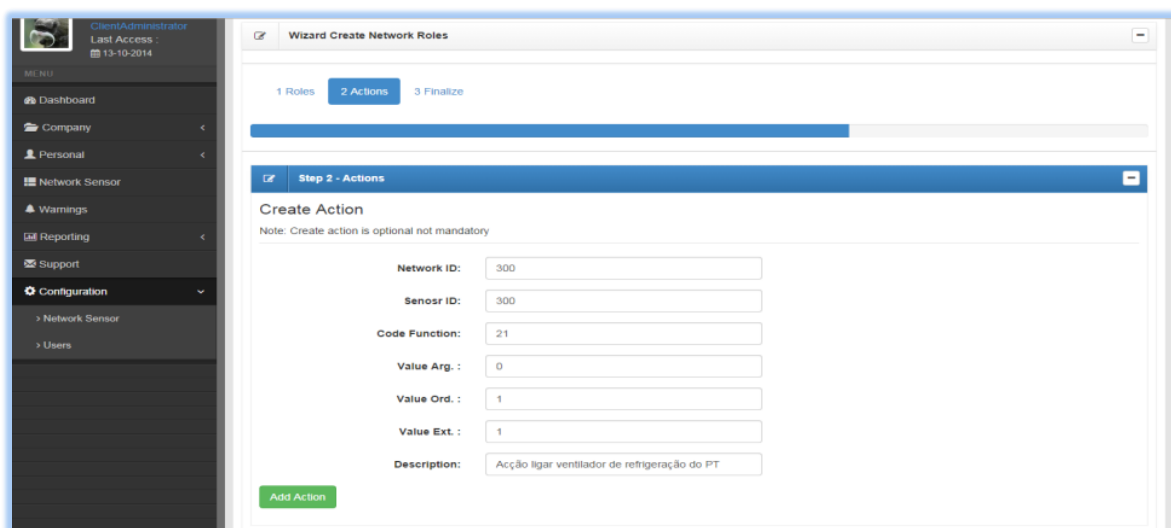
Periodic Condition: Minutes

Condition: >

Value Alarm: 40

Description: No periodo de um minuto foi superior 40 graus

Figura D-58 - Página Wizard para configurar regra



Client Administrator
Last Access : 13-10-2014

MENU

- Dashboard
- Company
- Personal
- Network Sensor
- Warnings
- Reporting
- Support
- Configuration
 - Network Sensor
 - Users

Wizard Create Network Roles

1 Roles 2 Actions 3 Finalize

Step 2 - Actions

Create Action

Note: Create action is optional not mandatory

Network ID: 300

Sensor ID: 300

Code Function: 21

Value Arg.: 0

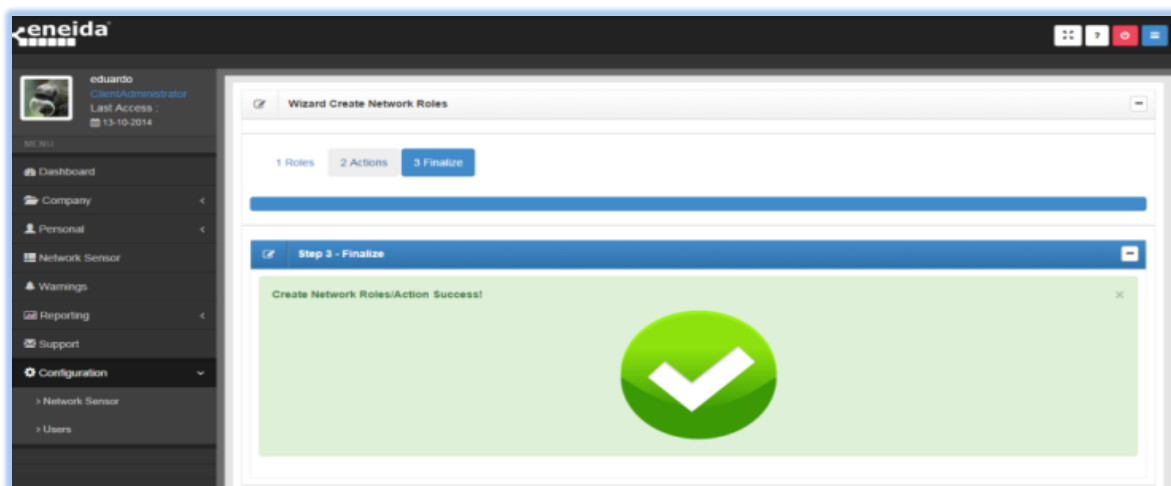
Value Ord.: 1

Value Ext.: 1

Description: Acção ligar ventilador de refrigeração do PT

Add Action

Figura D-59 - Página Wizard para configurar ação



eneida

eduardo
Client Administrator
Last Access : 13-10-2014

MENU

- Dashboard
- Company
- Personal
- Network Sensor
- Warnings
- Reporting
- Support
- Configuration
 - Network Sensor
 - Users

Wizard Create Network Roles

1 Roles 2 Actions 3 Finalize

Step 3 - Finalize

Create Network Roles/Action Success!

Figura D-60 - Página Wizard finalizada, regra para alarmes criado com sucesso

Na figura D-61, pode ver-se a lista de todas as regras e ações que o utilizador criou para as suas redes de sensores. Aqui o utilizador poderá eliminar as regras criadas ou então editar informação sobre cada regra criada.

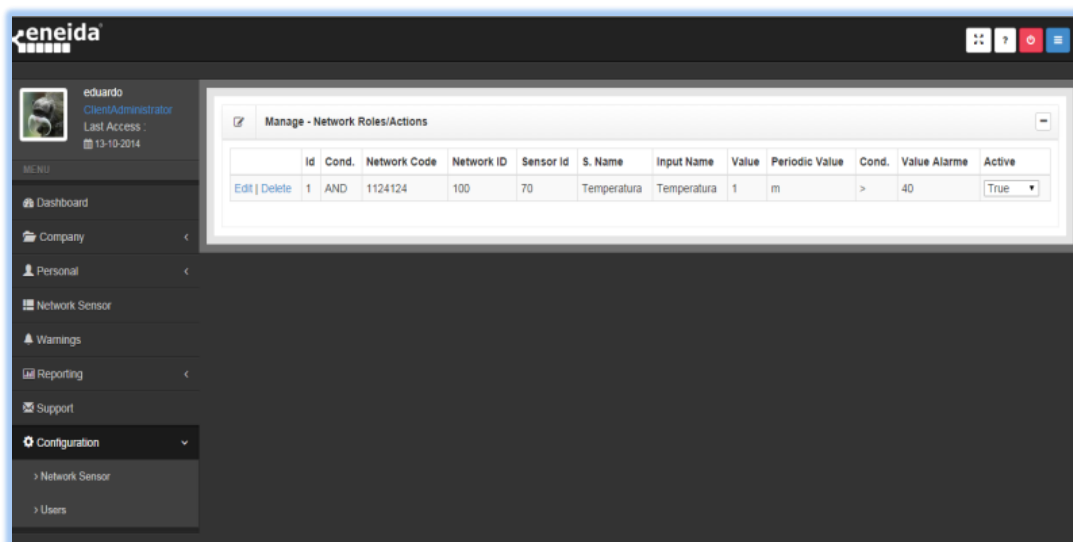


Figura D-61 - Página para gestão de regras para alarmes

Na figura D-62, é apresentada a página para configuração de todos os grupos e modelos para cada sensor. Aqui o utilizador poderá editar, listar e apagar informações relativas a grupos e modelos criados.

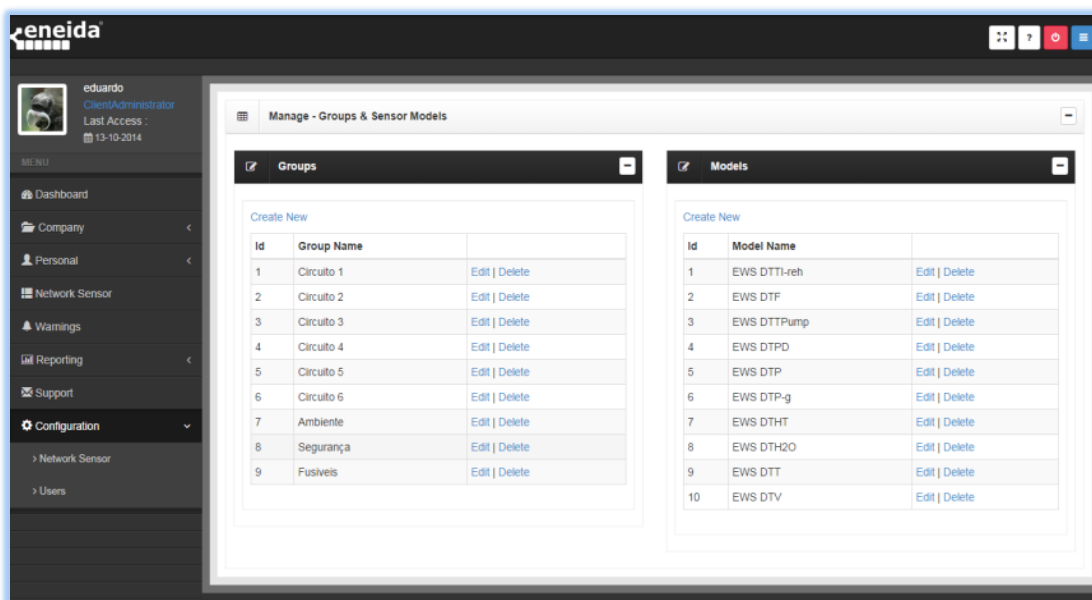


Figura D-62 - Página para configuração grupos e modelos

Página *Home*, opção do Menu *Administration*

A opção do menu *Administration* permite apenas ao utilizador *AdministradorGeral* aceder às seguintes opções: criar, editar, listar e apagar empresas, como se pode ver nas figuras D-63 e D-64; criar, editar, listar e apagar licenças para cada empresa criada, como se pode ver nas figuras D-65 e D-66, e criar, editar, listar e apagar utilizadores *ClientesAdministradores* para cada empresa criada, como se pode ver nas figuras D-67 e D-68.

Na figura D-63 é apresentada a página para criar empresa, e na figura D-64 é apresentada a página para listar todas as empresas criadas na aplicação *Web*.

Figura D-63 - Página para criar empresa na aplicação *Web*

Id	Country	Name	Vat	Email	Phone	Address	select
1	Portugal	Eneida, Wireless & Sensors, S.a	510097500000	sp@eneida.pt	+351 239 111 180	Ed. IPN, R...	Edit Delete
2	Portugal	Eneida, Wireless & Sensors, S.a	501.0007.00	sp@eneida.pt		1250-162 L...	Edit Delete
3	Portugal	Eneida, Wireless & Sensors, S.a	124125521	sp@eneida.pt		3030	Edit Delete
7	Portugal	Nome Empresa	193243434	email@empresa.pt	239234552	Rua Instit...	Edit Delete

Showing 1 to 4 of 4 entries

Figura D-64 - Página para listar empresas existentes na aplicação *SCADA*

Na figura D-65 é apresentada a página para criar licença para uma empresa e na figura D-66 é apresentada a página para listar todas as licenças criadas para cada empresa criada na aplicação Web.

e-neida

administrator
GeneralAdministrator
Last Access : 14-10-2014

EDP - Energias de Portugal

MENU

- Dashboard
- Company
- Personal
- Network Sensor
- Warnings
- Reporting
- Support
- Configuration
- Administration
 - Company
 - Licenses
 - Clients
 - Users

Create Licence

Company: Nome Empresa

Type License: Full Licence

Initial Date: 01-07-2014

End Date: 01-07-2015

Version: 1.0

Description: Licença Teste Empresa

submit

Figura D-65 - Página para criar licença para empresa

e-neida

administrator
GeneralAdministrator
Last Access : 11-11-2014

EDP - Energias de Portugal

MENU

- Dashboard
- Company
- Personal
- Network Sensor
- Warnings
- Reporting
- Support
- Configuration
- Administration
 - Company
 - Licenses
 - Clients
 - Users

Manage Licenses

10 records per page

Search: [] Copy CSV Excel PDF Print

ID	Company	Type License	start date	end date	version	description	visible	select
2	EDP - Energias de Portugal	Full Licence	09-06-2014 00:00:00	09-06-2015 00:00:00	1.0	Full Licence	True	Edit Delete
3	EDP - Energias de Portugal	Full Licence	09-06-2014 00:00:00	09-06-2014 00:00:00	1.0	Full Licence	True	Edit Delete
12	Nome Empresa	Full Licence	01-07-2014 00:00:00	01-07-2015 00:00:00	1.0	Licença Teste Empresa	True	Edit Delete

Showing 1 to 3 of 3 entries

Previous 1 Next

Figura D-66 - Página para listar licenças das empresas existentes na aplicação SCADA

Na figura D-67 é apresentada a página para criar utilizadores, e na figura D-68 é apresentada a página para listar todos os utilizadores de uma empresa existente na aplicação *Web*.

eneida

administrator
GeneralAdministrator
Last Access : 14-10-2014

Dashboard

Company

Personal

Network Sensor

Warnings

Reporting

Support

Configuration

Administration

Company

Licenses

Clients

Users

Create User

Name: Pedro Neves

Username: pedro_neves

E-mail: pedro_neves@empresa.pt

Role: ClientAdministrator

Company: Nome Empresa

submit

Figura D-67 - Página para criar utilizadores para uma empresa

eneida

eduardo
ClientAdministrator
Last Access : 13-10-2014

Dashboard

Company

Personal

Network Sensor

Warnings

Reporting

Support

Configuration

Network Sensor

Users

Manage Users

10 records per page

Search:

Copy CSV Excel PDF Print

Add Alert	Userid	UserName	Email	Name	Company	Date	Role	Select
active	2	eduardo	eduardo.net@sapo.pt	Eduardo A...	EDD - Energias de Portugal	13-10-2014	ClientAdministrator	Edit Delete
active	22	marta	speedynet13@gmail.com	Marta Neve...	EDD - Energias de Portugal	24-07-2014	ClientUser	Edit Delete

Showing 1 to 2 of 2 entries

Previous 1 Next

Figura D-68 - Página para listar utilizadores de uma empresa

ANEXO E

Testes

1. TESTES

Nesta secção apresentam-se os testes realizados às aplicações desenvolvidas para o novo sistema *SCADA* da *eneidaws*. Estes testes foram realizados durante o desenvolvimento das aplicações em ambientes de testes. Por falta de tempo e disponibilidade da empresa, não foi possível realizar testes em ambiente de produção. No entanto, foram feitos testes funcionais a todas as aplicações e testes de capacidade à aplicação servidor e à aplicação *Web Service*. Para fazer testes de capacidade à aplicação servidor e à aplicação *Web Service*, foi desenvolvida uma aplicação para emular as mensagens enviadas por *socket* e uma aplicação para emular mensagens por *POST* referentes às redes de sensores.

Apesar de não existir a possibilidade de fazer testes às aplicações em produção, foi possível importar os dados reais da base de dados da atual aplicação *SCADA DSI*, para a base de dados do novo sistema *SCADA*, tornando-se desta forma possível testar as funcionalidades das aplicações com dados reais do posto de transformação do caso de estudo da empresa. A cópia dos dados foi feita com a ferramenta *Talend Open Studio* para *data integration*.

Globalmente, o objetivo foi verificar que as aplicações funcionavam de acordo com o pretendido.













1.1. APLICAÇÃO SCADA WEB



Foram realizados os testes funcionais com o objetivo de validar as funcionalidades do sistema através da interação com a interface gráfica da aplicação e verificar os resultados esperados em cada teste. Na tabela E-1 são apresentados os testes realizados à aplicação *SCADA Web*. Os testes que não foram bem-sucedidos foram corrigidos, sendo apresentada a descrição de cada um.

Tabela E-1 - Testes Funcionais aplicação SCADA Web

Código Teste ¹	Descrição do Teste	Passou	Não Passou
T1ASW	"Fazer Login" – Validar que o utilizador é obrigado a inserir o seu <i>login</i> e <i>password</i> , para poder entrar na aplicação.	✓	
T2ASW	"Recuperar Password" - Validar que o utilizador pode recuperar a <i>password</i> , sempre que o mesmo tenha uma conta de acesso válida por <i>e-mail</i> .	✓	

¹ T1ASW – Teste número 1, Aplicação *SCADA Web*

T3ASW	“Ver Contactos” – Validar que o utilizador pode ver os contactos da empresa <i>eneidaws</i> .		
T4ASW	“Dashboard” – Validar que existem notificações dos alarmes críticos, alertas e os <i>links</i> para as outras opções da aplicação. Validar que os <i>links</i> das opções ficam visíveis de acordo com as permissões de cada utilizador, Cliente Utilizador, Cliente Administrador e Administrador Geral.		 *1
T5ASW	“Empresa” – Validar que todos os dados da empresa estão visíveis para o utilizador Cliente Administrador.		
T6ASW	“Informação Pessoal” – Validar que todos os dados referentes ao utilizador logado no sistema estão disponíveis.		
T7ASW	“Rede Sensores” - Validar que toda a informação para visualização das redes de sensores estão num mapa por georreferenciação e numa tabela. Validar que pode ser escolhida a visualização dos últimos dados adquiridos numa determinada rede, sabendo o estado de cada sensor. Esta opção está disponível para os utilizadores Cliente Utilizador, Cliente Administrador e Administrador Geral.		
T8ASW	“Alarmes” – Validar que é possível consultar toda a informação dos alarmes críticos e alertas adquiridos e as informações associadas aos mesmos. Validar que está disponível para os utilizadores Cliente Utilizador, Cliente Administrador e Administrador Geral. Validar que o utilizador Cliente Administrador pode validar alarme como tratado e colocar uma descrição, após verificar a sua ocorrência.		 *2
T9ASW	“Reporte” – Validar que é possível consultar o histórico dos dados adquiridos em gráficos, tabelas, e a possibilidade de exportar estas informações para formato Excel, <i>PDF</i> , entre outros. Validar igualmente a possibilidade de fazer a pesquisa de informação específica de sensores numa rede.		
T10ASW	“Suporte” – Validar a possibilidade de o utilizador Cliente Utilizador tirar dúvidas relativas à aplicação <i>web</i> com o utilizador Cliente Administrador. Validar também que o utilizador Cliente Administrador deve ter a possibilidade de tirar dúvidas com o ator Administrador Geral.		
T11ASW	“Configurações” – Validar a possibilidade de o utilizador Cliente Administrador poder: criar e gerir redes de sensores; criar e gerir regras para alarmes e associar utilizadores que recebem alarmes; criar e gerir os seus utilizadores e atribuir níveis de privilégios; importar e exportar redes de sensores automaticamente; visualizar erros ocorridos na configuração da rede de sensores.		
T12ASW	“Administração” – Validar que o utilizador Administrador Geral pode: criar e gerir empresas; criar e gerir licenças atribuídas a uma empresa; criar e gerir utilizadores associados a uma empresa.		

T13ASW	“Consultar Ajuda” – Validar que a funcionalidade consultar o manual de ajuda na aplicação está disponível.		
T14ASW	“Terminar Sessão” – Validar que o utilizador pode fazer <i>logout</i> , terminando a sessão e saindo da aplicação.		

*1 (T4ASW) – Após o utilizador fazer *login*, não estavam a ser validadas as permissões de acesso à aplicação de cada utilizador, ficando todas as opções disponíveis. Para solucionar este problema, foi necessário, após o login, serem validadas as permissões do utilizador.



*2 (T8ASW) – Ao fazer-se a listagem dos alarmes com a informação dos vários géneros de alarmes, estes estavam a aparecer misturados. Para resolver este problema, foi necessário refazer a consulta à base de dados que retornava a informação sobre as redes de sensores.

De modo geral, os testes indicaram que a aplicação está preparada para funcionar de acordo com o pretendido, realizando a configuração e a supervisão e controlo das diferentes redes de sensores.





1.2. APLICAÇÃO SERVIÇO DE ALARMES

Foram realizados os testes funcionais com o objetivo de validar as funcionalidades do sistema através da interação com a interface gráfica da aplicação e verificar os resultados esperados em cada teste. Na tabela E-2 são apresentados os testes realizados à aplicação serviço de alarmes. Os testes que não foram bem-sucedidos foram corrigidos, sendo apresentada a descrição de cada um.

Tabela E-2 - Testes Funcionais aplicação serviço de alarmes

Código Teste ²	Descrição do Teste	Passou	Não Passou
T1ASA	“Ligar Serviço” - Validar que o administrador geral é obrigado a configurar o relógio para verificação de alarmes em segundos, minutos ou dias. Garantir que o administrador não consegue ligar a aplicação serviço de alarmes caso não tenha configurado o mesmo.		 *1

² T1ASA – Teste número 1, Aplicação Serviço de Alarmes

T2ASA	“ <i>Verificar Alarmes</i> ” - Validar que a aplicação faz a verificação da existência de alarmes periodicamente, com base no relógio configurado, e caso seja encontrado alarme é notificado o utilizador por e-mail e/ou SMS.		
T3ASA	“ <i>Enviar e-mail e/ou SMS</i> ” - Validar que cada vez que existirem alarmes, estes devem ser enviados por SMS e/ou e-mail.	 *2	
T4ASA	“ <i>Desligar Serviço Alarmes</i> ” - Validar que, sendo desligado o serviço de alarmes, devem ser terminados todos os processos para verificação de alarmes.		
T5ASA	“ <i>Consultar Ajuda</i> ” – Validar que a funcionalidade consultar o manual de ajuda na aplicação está disponível.		

*1 (T1ASA) – Foi necessário fazer verificações para obrigar o utilizador a definir um relógio antes de iniciar a aplicação.

*2 (T3ASA) – Relativamente ao requisito não funcional com o código RNF2ASA, onde era pretendido envio de um *e-mail* e *SMS*, por cada alarme ocorrido no tempo máximo de 60 segundos, os testes indicaram que o período de envio nunca excede os 60 segundos para envio dos mesmos.

De modo geral, os testes indicaram que a aplicação está preparada para funcionar de acordo com o pretendido, realizando a verificação de alarmes no novo sistema *SCADA* e notificando o utilizador sempre que os mesmos ocorram.





1.3. APLICAÇÃO WEB SERVICE

Foram realizados os testes funcionais com o objetivo de validar as funcionalidades do sistema através da interação com a interface gráfica da aplicação e verificar os resultados esperados em cada teste. Na tabela E-3 são apresentados os testes realizados à aplicação *Web Service*. Os testes que não foram bem-sucedidos foram corrigidos, sendo apresentada a descrição de cada um.

Tabela E-3 - Testes Funcionais aplicação Web Service

Código Teste ³	Descrição do Teste	Passou	Não Passou
T1AWS	“Valida Chave Web Service” – Validar que a chave única a ser utilizada por cada empresa permite o acesso a informação do Web Service.	✓	
T2AWS	“Obter todas as localizações” - Validar que é disponibilizada toda a informação em JSON, relativamente a todas as localizações.	✓	
T3AWS	“Obter todas as Redes” - Validar que é disponibilizada toda a informação em JSON, relativamente a todas as redes.	✓	
T4AWS	“Obter todos os Sensores” - Validar que é disponibilizada toda a informação em JSON, relativamente a todos os sensores.	✓	
T5AWS	“Obter todas Redes de uma localização” - Validar que é disponibilizada toda a informação em JSON, relativamente a todas as redes de uma localização.	✓	
T6AWS	“Obter todos os Grupos” - Validar que é disponibilizada toda a informação em JSON, relativamente a todos os grupos.	✓	
T7AWS	“Obter todos os Sensores de um Grupo” - Validar que é disponibilizada toda a informação em JSON, relativamente a todos os sensores de um grupo.	✓	
T8AWS	“Obter todos os Modelos” - Validar que é disponibilizada toda a informação em JSON, relativamente a todos os modelos.	✓	
T9AWS	“Obter todos os Sensores de um Modelo” - Validar que é disponibilizada toda a informação em JSON, relativamente a todos os sensores de um modelo.	✓	
T10AWS	“Obter todas as Entradas de um Sensor” - Validar que é disponibilizada toda a informação em JSON, relativamente a todas as entradas de um sensor.	✓	
T11AWS	“Obter todas as Mensagens de uma Entrada” - Validar que é disponibilizada toda a informação em JSON, relativamente a todas as mensagens de dados de uma entrada.	✓	
T12AWS	“Obter todas as Mensagens” - Validar que é disponibilizada toda a informação em JSON, relativamente a todas as mensagens de dados.	✓	
T13AWS	“Obter intervalo de Mensagens” - Validar que é disponibilizada toda a informação em JSON, relativamente a todas as mensagens de um intervalo (por exemplo, as 15 últimas mensagens recebidas).	✓	
T14AWS	“Submete Mensagem” - Validar que é recebida toda a informação em JSON, de uma determinada mensagem relativa à rede de sensores.	✓	✗ *1
T15AWS	“Tratar Mensagens” - Validar que o Web Service é capaz de receber mensagens da rede de sensores.	✓	

³ T1AWS – Teste número 1, Aplicação Web Service

T16AWS	“ <i>Guardar Mensagem Base de Dados</i> ” - Validar que o <i>Web Service</i> é capaz de guardar todas as mensagens recebidas em base de dados.		 *2
T17AWS	“ <i>Verificar Alarmes</i> ” - Validar que é possível ativar ou desativar verificação de alarmes. Se a opção estiver ativa, todas as mensagens recebidas pela rede de sensores devem ser validadas relativamente há existência de alarmes. Se a opção estiver desativada, não devem ser verificados os alarmes.		
T18AWS	“ <i>Enviar e-mail e/ou SMS</i> ” - Validar que, estando a opção verificar alarmes ativa, cada vez que existir um alarme, devem ser enviados notificações de alarmes por <i>SMS</i> e/ou <i>e-mail</i> .		

*1 (T14AWS) – No teste inicial verificou-se um problema no método para receção de informação no *POST* em *JSON*, dado ser necessário receber e ter como argumento a chave única de acesso ao *Web Service*. Após corrigido este problema, a receção de mensagens tornou-se possível.

*2 (T16AWS) – Após a receção de mensagens, o método que recebe a informação, no momento de guardar a informação na base de dados, não a guardava na respetiva base de dados do cliente. Foi necessário fazer a verificação de a que empresa pertencia a mensagem e depois guardar a informação na base de dados.










De modo geral, os testes indicaram que a aplicação está preparada para funcionar de acordo com o pretendido, realizando-se a verificação e disponibilização de informação relativa à rede de sensores. Em particular, a empresa só pretendia métodos para disponibilização de informação. No entanto, o aluno considerou que o custo de implementação de um método para submissão de dados seria baixo e, no futuro, se a *eneidaws* pretender adotar outro modo de comunicação com a rede de sensores, poderá desenvolver *hardware* capaz de submeter dados através de *POST* para o *Web Service* em formato *JSON*.

Para demonstrar como esta poderá ser uma alternativa, foram realizados testes de carga na submissão de dados através de *POST* no formato *JSON*, para o *Web Service*.










1.4. APLICAÇÃO SERVIDOR

Foram realizados os testes funcionais com o objetivo de validar as funcionalidades do sistema através da interação com a interface gráfica da aplicação e verificar os resultados esperados em cada teste. Na tabela E-4 são apresentados os testes realizados à aplicação servidor. Os testes que não foram bem-sucedidos foram corrigidos, sendo apresentada a descrição de cada um.

Tabela E-4 - Testes Funcionais aplicação servidor

Código Teste ⁴	Descrição do Teste	Passou	Não Passou
T1AS	“Ligar Servidor” – Validar que o administrador geral é obrigado a configurar o IP e porto do servidor, antes que as comunicações fiquem ativas. Garantir que o administrador não consegue ligar a aplicação servidor caso não tenha configurado o IP e porto.		 *1
T2AS	“Submete Mensagem” – Validar que o servidor é capaz de tratar todas as mensagens recebidas pela rede de sensores.		
T3AS	“Submete Ação Mensagem” – Validar que a aplicação web, ao querer executar uma ação numa rede de sensores, submete uma mensagem para o servidor. Validar que o servidor é capaz de tratar todas as mensagens recebidas pela aplicação web, encaminhando esta mensagem/ação para a rede de sensores.		
T4AS	“Tratar Mensagens”- Validar que o servidor é capaz de receber e enviar todas as mensagens.		
T5AS	“Guardar Mensagem Base de Dados” - Validar que o servidor é capaz de guardar as mensagens recebidas e enviadas pelo servidor nas bases de dados correspondentes a cada cliente da eneidaws.		 *2
T6AS	“Verificar Alarmes” – Validar que o utilizador pode opcionalmente ativar ou desativar a verificação de alarmes. Se a opção estiver ativa, validar que todas as mensagens recebidas pela rede de sensores devem ser validadas quanto à existência de alarmes. Se a opção estiver desativada, validar que não devem ser verificados alarmes.		
T7AS	“Enviar e-mail ou SMS” – Validar que, estando a opção verificar alarmes ativa, cada vez que existir um alarme devem ser enviados alarmes por SMS e/ou e-mail. Se a opção verificar alarmes estiver desativada, validar que não são enviados alarmes por e-mail e/ou SMS.		

⁴ T1AS – Teste número 1, Aplicação Servidor

T8AS	“Desligar Servidor” – Validar que, ao ser desligado o servidor, todos os processos de comunicação que estejam a decorrer devem ser terminados.		
T9AS	“Escolher Opção” – Validar que o utilizador pode listar todas as opções que pretenda escolher.		
T10AS	“Enviar Alive ID” – Validar que o utilizador pode enviar mensagens às redes conectadas ao servidor, “perguntando” quais são os equipamentos que se encontram ativos nessas redes.		*3
T11AS	“Enviar Mensagem Teste” - Validar que o utilizador pode enviar mensagens de teste para um equipamento numa rede.		*4
T12AS	“Listar Clientes Conectados” - Validar que o utilizador pode listar todas as redes ligadas ao servidor.		
T13AS	“Ligar Log” – Validar que o utilizador pode ligar o Log que permite visualizar as mensagens que estão a ser trocadas com o servidor.		
T14AS	“Desligar Log” - Validar que o utilizador pode desligar o Log que permite visualizar as mensagens que estão a ser trocadas com o servidor.		
T15AS	“Limpar Log” - Validar que o utilizador pode limpar o Log que permite visualizar as mensagens que estão a ser trocadas com o servidor.		
T16AS	“Consultar Ajuda” – Validar que a funcionalidade consultar o manual de ajuda na aplicação está disponível.		

*1 (T1AS) – Foi necessário fazer verificações para obrigar o utilizador a definir um *IP* e porto antes de iniciar a aplicação. Quando o utilizador mandava iniciar a aplicação sem definir *IP* e porto a aplicação dava erro.

*2 (T5AS) – Problema: quando era recebida uma mensagem de um determinado cliente, a informação não era guardada na base de dados desse cliente. Foi necessário implementar um método que permitisse alterar em *runtime* a conexão à base de dados de cada cliente.

*3 (T10AS) *4 (T11AS) – Estas mensagens de *alive id* foram testadas pelo programa de captura de mensagens denominado *hércules*⁵, dado não se ter disponível *hardware* da *eneidaws* para realizar estes testes.

⁵ Informação do programa *hércules*. Disponível no endereço: http://www.hw-group.com/products/hercules/index_en.html

De modo geral, os testes indicaram que a aplicação está preparada para funcionar de acordo com o pretendido, realizando a recepção de mensagens da rede de sensores e o armazenamento dos dados das redes em base de dados. Foram realizados testes de capacidade na recepção de mensagens das redes de sensores.

1.5. TESTES DE CAPACIDADE À APLICAÇÃO SERVIDOR E *WEB SERVICE*

Os testes de capacidade realizados à aplicação servidor e à aplicação *Web Service* foram feitos de igual forma, no entanto o método de comunicação em cada uma é diferente. Na aplicação servidor as comunicações são feitas através de ligações *socket* por *TCP/IP*, e na aplicação *Web Service* as comunicações são feitas por *POST* via *HTTP* no formato *JSON*.

Para testar estas duas aplicações foi necessário desenvolver duas outras aplicações que permitissem simular as mensagens das redes de sensores nos dois tipos de comunicações.

O cenário implementado para realizar testes de capacidade às comunicações das aplicações servidor e *Web Service* foi efetuado numa rede doméstica, em casa do aluno, como se pode ver na figura E-1. Usaram-se cinco computadores para fazer submissão de mensagens simulando os dados provenientes das redes de sensores para o computador central, onde se encontram alojadas todas as aplicações do novo sistema *SCADA*.

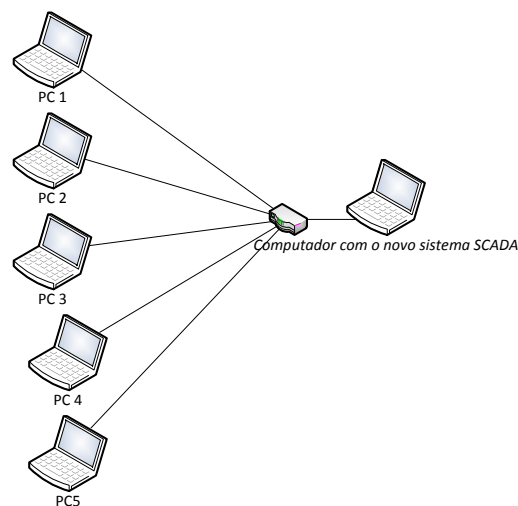


Figura E-1 - Exemplo do cenário usado para realizar os testes às comunicações

Numa rede de sensores podem existir vários sensores a debitar dados para a aplicação servidor. Por exemplo, nos testes realizados constatou-se que o tempo da recepção de 1000 mensagens enviadas por um único computador ou a recepção do mesmo número de mensagens enviadas por vários computadores é semelhante.

Nos testes realizados utilizaram-se vários computadores com diversos débitos de mensagens, como se pode ver na tabela E-5, onde se simulou a probabilidade de vários sensores de uma rede, ou vários sensores de várias redes, reportarem os seus dados num mesmo instante e o respetivo tempo de processamento destas mensagens. Como se pode ver no gráfico da figura E-2, onde são apresentados os resultados dos vários testes, à medida que o número de mensagens aumenta, o tempo de processamento das mensagens cresce exponencialmente.

Tabela E-5 – Intervalos utilizados para envio de mensagens

Nº de Mensagens (Enviadas)	Tempo de Processamento (Total Segundos)	Tempo de Processamento (Segundos)	Frequência (Mensagens/T. Segundos)
10	3 Segundos	0,3 s	3,33
100	7 Segundos	0,07 s	14,28
1000	48 Segundos	0,048 s	20,83
10.000	468 Segundos	0,0468 s	21,37
100.000	4440 Segundos	0,0444 s	22,52

O comportamento no tempo de processamento das mensagens recebidas através de *socket* por *TCP/IP* pela aplicação servidor e o tempo de processamento das mensagens recebidas por *POST* via *HTTP* pela aplicação *Web Service* é semelhante, como se pode constatar no gráfico da figura E-3. Por esta razão, o aluno considerou que seria uma mais-valia para a empresa, no futuro, existir uma outra alternativa para as redes de sensores terem outro meio de comunicação para reportar os seus dados, e o *Web Service* poderá constituir esta alternativa.

O uso do *Web Service* poderá ter vantagem relativamente à comunicação por *sockets*, tendo em conta o seguinte exemplo: se existirem, por exemplo, 1000 valores a serem enviados pela rede de sensores através de *socket* por *TCP/IP*, com o atual protocolo de comunicações usado pela *eneidaws* será necessário submeter 1000 mensagens uma a uma para o servidor, consumindo-se o tempo de 48 segundos, nas condições iguais às realizadas nos testes.

Caso a *eneidaws* possua *hardware* capaz de produzir *POST* e submeter valores no formato *JSON*, os 1000 valores poderão ser todos enviados numa única mensagem em vez de comunicar os valores um a um, ganhando-se tempo no processo de comunicação por não ser necessário esperar pelo tempo de confirmação. No entanto, o custo do *hardware* capaz de construir a mensagem e submeter dados neste formato poderá ser um pouco mais elevado. Caberá à *eneidaws* analisar e decidir se esta pode ser uma alternativa válida a ser adotada nas suas comunicações. Considera-se uma mais-valia deste projeto propor esta alternativa à *eneidaws*.

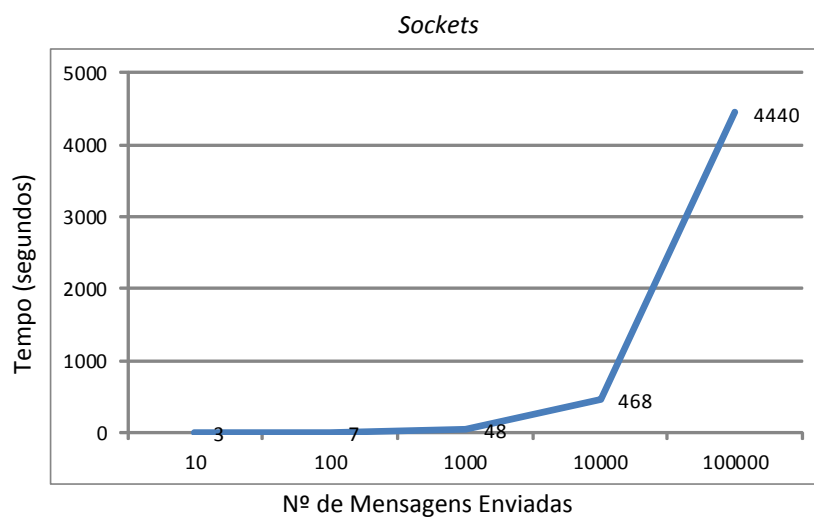


Figura E-2 - Gráfico recepção de dados das redes de sensores por socket para aplicação servidor

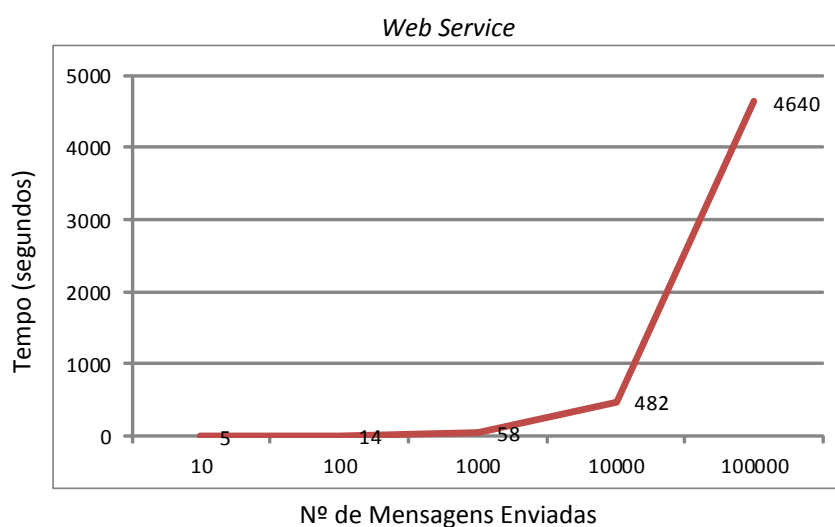


Figura E-3 - Gráfico recepção de dados das redes de sensores por POST para Web Service

Nos testes realizados é importante considerar os seguintes pontos: os valores obtidos no processamento das mensagens superam largamente os valores definidos como mínimos pela *eneidaws* (tempo de processamento de cada mensagem nunca superior a 60 segundos), sendo apresentado o pior valor dos testes 0,3 segundos por mensagem. É necessário ter em conta que estamos numa rede doméstica em que o tempo de transmissão no envio de mensagens nunca é superior 10 milissegundos, não criando muito atraso nas comunicações. Nas várias redes que a *eneidaws* venha a projetar, os valores correspondentes podem alterar-se mediante a tipologia da rede e o tipo de comunicações (por exemplo fibra, *GPRS*, *3G*, *4G*, etc.).

Outro aspeto a ter em consideração será os computadores que foram usados serem de gama doméstica com baixa capacidade de memória e processamento. Prevê-se que com *hardware* de gama profissional, como servidores com grande capacidade de memória e processamento, poderão existir melhores resultados devido a estas características. Infelizmente, não houve a possibilidade de usar este género de *hardware* nos testes.

Pela análise dos gráficos pode-se constatar que, a partir das 10.000 mensagens de dados no mesmo instante, o tempo de processamento aumenta exponencialmente, pelo que se considera que a aplicação servidor e a aplicação *Web Service*, nestas condições, têm um tempo máximo de processamento aceitável entre as 10.000 e 50.000 mensagens com valores reportados em simultâneo. Considera-se que, em condições reais, este volume de mensagens no mesmo instante será pouco provável de ocorrer. Por exemplo, no caso de estudo da empresa, cada posto de transformação tem em média 20 sensores a reportar dados de hora-a-hora ou minuto-a-minuto. Não existe a necessidade de uma maior frequência de valores nas comunicações por parte do cliente da *eneidaws*.

No entanto, pressupondo uma situação em que 500 postos de transformação representam 500 redes de sensores com uma média de 20 sensores por rede, consegue-se, numa situação limite, ter 10.000 mensagens no mesmo instante com um tempo de processamento de cerca de 468 segundos, em condições semelhantes às dos testes, o que representa um tempo aceitável de acordo com os requisitos mínimos exigidos pela empresa. Se, porventura, a empresa pretender melhores tempos de processamento nas várias redes que venha a implementar, poderá utilizar várias aplicações servidor e aplicações *Web Service*, sendo que cada uma terá um determinado número de redes de sensores a seu cargo, conseguindo-se desta forma um melhor balanceamento de carga no processamento das mensagens. Como alternativa, a *eneidaws* poderá ainda usar diferentes máquinas físicas com várias aplicações servidor e aplicações *Web Service* para melhorar o desempenho no processamento das mensagens recebidas das redes de sensores.

Na globalidade, os testes realizados às aplicações cumprem todos os requisitos definidos (requisitos funcionais e requisitos não funcionais), tendo estes sido executados em ambiente de desenvolvimento, uma vez que não foi possível realizar testes em produção. Considera-se que o novo sistema *SCADA* atinge todos os objetivos definidos inicialmente, tal como pretendido pela empresa *eneidaws*.

